NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

VERIFICATION AND VALIDATION OF THE MALICIOUS ACTIVITY SIMULATION TOOL (MAST) FOR NETWORK ADMINISTRATOR TRAINING AND EVALUATION

by

Justin M. Neff

March 2012

Thesis Co-Advisors: Gurminder Singh
John H. Gibson

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE | *Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2012	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE Verification and Validation of the Malicious Activity Simulation Tool (MAST) for Network Administrator Training and Evaluation

5. FUNDING NUMBERS

6. AUTHOR(S) Justin M. Neff

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ___N/A________.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

The DoD currently employs red teams to conduct network infiltration and security training for network administrators and operators. Red Teams provide the most realistic and thorough training to defend against real-world threats and we are developing a system to mimic this highly trained adversary based on the proof of concept framework developed by CDR Will Taff and LCDR Paul Salevski.

This thesis is meant to perform a verification and validation analysis of the suitability of the MALWARE Mimic concept as a methodology for conducting network administrator network security training and awareness, alleviation of red team availability constraints, and network user security awareness training. We also develop a strategy by which the effectiveness of the MALWARE Mimic system for increasing such network security awareness and elevating the information assurance posture of distributed command networks can be measured.

14. SUBJECT TERMS Red Team, Malware, Network Administrator Training, Computer Network Defense

15. NUMBER OF PAGES
111

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

VERIFICATION AND VALIDATION OF THE MALICIOUS ACTIVITY SIMULATION TOOL (MAST) FOR NETWORK ADMINISTRATOR TRAINING AND EVALUATION

Justin M. Neff
Lieutenant, United States Navy
B.S., Old Dominion University, 2006

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2012

Author: Justin M. Neff

Approved by: Gurminder Singh
Thesis Co-Advisor

John H. Gibson
Thesis Co-Advisor

Peter J. Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The DoD currently employs red teams to conduct network infiltration and security training for network administrators and operators. Red Teams provide the most realistic and thorough training to defend against real-world threats and we are developing a system to mimic this highly trained adversary based on the proof of concept framework developed by CDR Will Taff and LCDR Paul Salevski.

This thesis is meant to perform a verification and validation analysis of the suitability of the MALWARE Mimic concept as a methodology for conducting network administrator network security training and awareness, alleviation of red team availability constraints, and network user security awareness training. We also develop a strategy by which the effectiveness of the MALWARE Mimic system for increasing such network security awareness and elevating the information assurance posture of distributed command networks can be measured.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ABM	Asset Baseline Monitor
CDX	Cyber Defense Exercise
COMPTUEX	Composite Training Unit Exercise
CSTT	Combat Systems Training Team
DCM	Device Control Module
DDoS	Distributed Denial of Service
DNS	Domain Name System
DoD	Department of Defense
ePO	ePolicy Orchestrator
GB	Giga Byte
GHz	Giga Hertz
HBSS	Host Based Security System
HIPS	Host Intrusion Prevention System
HTTP	Hyper Text Transfer Protocol
IA	Information Assurance
IAVA	Information Assurance Vulnerability Alert
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
IT	Information Technology
MAST	Malicious Activity Simulation Trainer
NCDOC	Navy Cyber Defense Operational Center
NSA	National Security Agency
NSST	Navigation Seamanship and Ship-handling Trainer
OPFOR	Opposing Forces
OSI	Open Systems Interconnection
PA	Policy Auditor
RAM	Random Access Memory
TB	Terabyte

TCP	Transmission Control Protocol
TTO	Training Time Out
TTP	Tactics, Techniques and Procedures
VM	Virtual Machine

ACKNOWLEDGMENTS

This thesis would not have been possible without the guidance and patience of my thesis advisor, Professor Singh. Thank you for imparting your experience and insight and for your continued support throughout the process. My thesis co-advisor, Mr. John Gibson, provided tireless support and guidance throughout the thesis and for that I thank you.

I would be remiss if I did not offer my sincere thanks to CDR Jim Hammond, and Capt Ray Longoria, my “partners in crime” on this project. Also, I would like to thank Mr. Arijit Das, Mr. Erik Lowney, and Mr. Greg Belli for their contributions to the project and assistance with this thesis. Additionally, I want to thank Ms. Susan Hood from SPAWARSYSCEN-PACIFIC and Mr. Quincy Taitt from Man Tech systems for providing the software for our implementation platform as well as training on HBSS.

Finally, to my wife, Tara, I thank you for your continued support and for taking care of everything on the home front while I worked on this thesis. You are the quintessential Navy Wife and, as always, you have my love and admiration. To my children, Jayden, Preston, and Carson, you have my unwavering love and thank you for making life so interesting and so much fun!

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The Department of Defense (DoD) continues to be increasingly reliant on information technology and the associated networked infrastructure to complete its various missions. The automation of some critical tasks has helped enabled the DoD to meet the challenge of protecting the United States of America; however, the cyber domain is a rapidly evolving environment with its own inherent threats and security challenges. Malicious software in the form of Internet worms, viruses, and botnets, as well as other threats, pose a great security risk to the DoD and ipso facto the security of the nation. To counteract the continuously evolving cyber security threats, the DoD conducts training for network administrators and operators to raise their awareness of malicious software behaviors and to increase the Information Assurance readiness of DoD networks.

A. NETWORK ADMINISTRATOR TRAINING

To ensure that DoD network administrators are properly equipped with the skills necessary to defend their networks, the DoD conducts training through various methods. Classroom training is often utilized to familiarize network administrators with current security threats and how to handle them. Classroom network laboratories are also utilized to conduct “hands-on” training with malicious software and network vulnerabilities. However, the most realistic network security training is conducted with the use of red teams [1], [2]. Red teams are groups of highly skilled personnel that act as adversaries to test network administrators on the recognition of malware and current vulnerabilities that our networks are facing.

B. DEFICIENCIES IN OUR CURRENT APPROACH TO TRAINING

While red teams provide the most effective training currently, there are some inherent constraints associated with the use of red teams for training network administrators. With the ever increasing reliance on information systems to conduct all facets of missions in the DoD, the demand for training of network administrators using red teams has skyrocketed. Red teams are a constraining resource due to the advanced skill sets they possess, limited budgets, and increased operational tempo due to the increased demand; and for these reasons, the red teams simply cannot keep up with the demand to conduct training across the DoD.

C. OBJECTIVES

With the establishment of the United States Tenth Fleet/ Fleet Cyber Command for cyber warfare, and the cyber domain evolving as a warfare area, we need to develop a strategy for augmenting the current training structure which is dependent on the resource-constrained red teams. Towards this end, we seek to leverage the framework previously developed by CDR Will Taff and LCDR Paul Salevski and further their research of a software based "Malware Mimic" training tool to increase the standardization and availability of network cyber defense training [1].

In this thesis, we perform verification and validation analysis of the suitability of the Malware Mimic-based approach implemented in Malicious Activity Simulation Tool (MAST), for conducting network administrator network security training and awareness, alleviation of red team availability constraints, and network user security awareness training. Further, we develop a strategy for assessing the effectiveness of MAST for increasing such network security awareness and elevating the information assurance posture of distributed command. Based on application of this strategy, MAST provides an extensible, robust capability to assess network administrator and user security awareness and compliance.

D. ORGANIZATION

Chapter I provides a brief description of the problem statement as well as motivation for the research, i.e., the increased security of the DoD’s computer network assets and ipso facto, the security of the nation as a whole.

Chapter II outlines previous research conducted in this area and describes the implications of that research. Chapter II also provides a more formal definition of red teams and provides some examples of how red teams are utilized to conduct training in the DoD. Additionally, Chapter II discusses some of the threats and vulnerabilities that DoD network administrators currently face. It also discusses how we currently conduct network security testing and training with red teams and also further delineates some of the shortcomings of our current approach to training network administrators with red teams.

Finally, Chapter II presents an example training scenario with MAST to lay the foundation for verifying and validating the tool.

Chapter III discusses the design considerations and test platform implemented to further develop the MAST system and that must be validated as a training tool. Chapter III also provides an overview of the MAST system and how we intend it to be used as a training tool. Additionally, Chapter III describes the hardware and software that we utilized to implement the MAST system. Finally, we discuss the Host Based Security System (HBSS) that is currently deployed on DoD networks to provide security to Windows and Unix based servers and workstations.

Chapter IV provides a critical analysis of red teams and ethical hackers versus the MAST system by analyzing some common methods employed by red teams and discussing how the MAST system will accomplish similar tasks while reducing the risk associated with the training. Additionally, Chapter IV discusses some metrics for comparison of the MAST system with red teams and other network administrator training tools. Finally, Chapter IV discusses some methods for measuring the effectiveness of the MAST system in mimicking training conducted by red teams.

Chapter V provides conclusions as a result of this study. Additionally, Chapter V outlines future work to be conducted on this project before it is ready for final testing and production.

II. BACKGROUND

This chapter provides insight into what red teams are and how they are employed in the DoD to provide training to network administrators. Additionally, this chapter offers insight into typical approaches and threats that red teams utilize to infiltrate DoD networks. Finally, this chapter explores how we currently train and discusses some issues associated with our reliance on red teams.

A. RED TEAMS

A red team "seeks to behave in a manner consistent with the world view and cultural beliefs of a potential adversary" [2]. Red teams are typically comprised of specially selected individuals who are trained to anticipate and simulate the behaviors of potential adversaries in order to achieve the most realistic training. According to Committee on National Security Systems (CNSS) Instruction Number 4009 (National Information Assurance Glossary), a red team is defined as:

> A group of people authorized and organized to emulate a potential adversary's attack or exploitation capabilities against an enterprise's security posture. The Red Team's objective is to improve enterprise Information Assurance by demonstrating the impacts of successful attacks and by demonstrating what works for the defenders (i.e., the Blue Team) in an operational environment. [3]

The purpose of red teams is to "challenge the effectiveness of new operational concepts in future crisis and conflicts" [2]. An added enhancement to the training provided is the fact that red teams are not scripted opponents and that they are actually dynamic adversaries that adapt to the trainee’s actions and the scenario. The use of red teams in training forces an organization to look critically at its policies and procedures to ensure that it is up to the task of facing a live, thinking adversary.

The concept of a red team is not new; for years, our military has tested operational units’ Tactics, Techniques, and Procedures (TTPs) against Opposing Forces (OPFOR) with the intent of simulating a dynamic adversary. OPFOR and red teams are utilized at all levels of military planning and training. The armed forces currently utilize OPFOR at the tactical level, such as infantry and mechanized units at the Army’s National Training Center, Fort Irwin, CA, as well as aviation units at the Navy’s Aggressor Squadron at Naval Air Station, Fallon, Nevada, to train units against a live, dynamic adversary. The red teams, which are typically led by staff Intelligence Officers, have studied our potential adversaries and use this knowledge to create realistic combat training scenarios against a credible, dynamic opponent [2].

The global war game held annually at the Naval War College in Newport, Rhode Island, is an example of red teams being utilized at the strategic level in order to hone our TTPs against a real, thinking opponent. The use of red teams and the global war game have had a significant impact on national and military guidance publications, such as the National Security Strategy, Joint Strategy Review, and the Maritime Strategy. One of the most notable instances of this influence is that the global war game from 1984 to 1988 completely reshaped our way of naval thinking, which led to the development of the Maritime Strategy [4]. By utilizing red team think tanks and war gaming scenarios, the possible actions of forecasted adversaries can be identified. This information might then influence the policy and actions of the entire DoD.

With regard to cyber security, red teams are a vital component of training for government and military information systems operators. The term operator can be very broad and might include the entire staff, such as managers and officers, administrators, engineers, help desk, and technicians. For this thesis, the scope of the term operator will be limited to administrators, help desk personnel, and technicians.

One example of red teams being used during a cyber-security training exercise is the annual Cyber Defense Exercise (CDX) in which the United States Service Academies, as well as other military academic institutions (Air Force Institute of Technology, Naval Postgraduate School), square-off against the National Security Agency red team. In the weeks leading up to CDX, the competing schools build their networks from the ground-up and are required to meet a baseline of functional services to include a web service, a domain name service, active directory, e-mail, and bulletin board. The students must then research the security issues facing their systems and harden their systems accordingly with patches and other methods; the goal being to minimize security risks. After the students have their networks built and functional, they must then connect to the game network via a Virtual Private

Network, which is logically removed from the Internet. The culmination of all of the preparations is the CDX in which the competing schools' networks and students must endure numerous attacks from the NSA red team [1], [5].

According to the Certified Ethical Hacking Manual, there are five phases by which a hacker progresses the attack [6]. The five phases are: reconnaissance, scanning, gaining access, maintaining access, and covering their tracks. During a previous CDX, the NSA red team followed these phases although the reconnaissance phase was shortened, since the red team already had significant knowledge of the competing schools' networks. The NSA red team also did not cover their tracks so that the students would be able to recognize when their networks had been compromised. The students gain valuable experience on how to detect if they have been scanned, infiltrated, or exploited. Once the students detect that there networks have been infiltrated, they must take actions to neutralize the problem, take corrective actions to restore the system, and finally, conduct research and implement controls so that the vulnerability is not exploited again.

There is a great deal of training and benefit to be had from exercises like CDX. However, this type of training has a few drawbacks. First and foremost, this training is not conducted on an actual live network and, since the students are building their networks to withstand pending attacks, the majority of their design decisions are based on security, which is unrealistic for an actual functioning network that must serve hundreds or even thousands of users. Secondly, this type of training on an exercise network is not a feasible method of training network operators across the DoD as it is too time intensive and doesn’t entirely transfer to their operational networks. Additionally, there is a shortage of red teams to conduct the amount of training required for an enterprise such as the DoD [7]. The cyber security training value that is provided by red teams is highly valuable and is the level of training for which the DoD should strive.

An example of an exercise where red teams conduct training on operational networks is the Composite Training Unit Exercise (COMPTUEX). The COMPTUEX is the capstone training exercise for a strike group after which the ships are assessed for their readiness to deploy and engage in battle. COMPTUEX is a multifaceted exercise that is designed to stress the entire strike group, from the staff to the individual ship’s officers and crew, as well as the air detachment and the Marines (if embarked).

During COMPTUEX, the operational computer networks onboard the strike group ships are attacked by the red team from the Navy Cyber Defense Operational Center (NCDOC). This training is highly beneficial to the network operators since it is on their operational networks. The training that is provided is similar to that of CDX; however, the red teams are even further limited on the extent of the attacks that they can perform since some of the more nefarious attacks could have catastrophic consequences on the operational networks and the strike group’s level of readiness. The NCDOC red teams follow a similar theme as the CDX red teams by first scanning the ship’s networks, followed by infiltration, and finally, exploitation.

Throughout the exercise, there are various attacks and evolutions that must be completed by the whole strike group and there are also events specific to a ship or a subset of the group. Since these exercises need to be graded by the training teams, there are times when the red team will not strike so that the crew's performance can be judged against the other various conventional attacks. However, some cyberattacks during COMPTUEX are scripted to coincide with other threats such as an attack on critical communications systems that the strike group is facing. This allows the evaluators to answer questions such as, "Can the ship fight without its full complement of communications?" [1].

The training value is greatly increased by having the red teams attack the operational networks, since these are the networks with which the administrators and technicians are most familiar and feel most comfortable. Upon completion of the exercise, the IT personnel have a thorough understanding of their network, its vulnerabilities, what their sensors can detect, and how to recover from an attack. However, a significant drawback to this training is that the red teams are limited and may not use their full repertoire of attacks so that they do not cause damage to the operational network. This detracts from the realism of the training since an actual attacker would not be restricted in the types of exploits that could be used.

Red teams are comprised of "ethical hackers" which according to the Certified Ethical Hacking Manual are "hired by organizations to attack their information systems and networks in order to discover vulnerabilities and verify that security measures are functioning correctly” [6]. Ethical hackers are typically contracted to conduct security testing which allows them to legally hack a network for defensive and security purposes. Red teams use ethical hacking in order to test the network security of DoD assets and to provide feedback on the state of their security.

Exercises incorporating a red team provide the most realistic training available. Red teams are utilized in the capstone exercises to certify the readiness of all deploying forces. As discussed previously, red teams can attack from a variety of angles. In this thesis, we will examine a specific subset of attacks that present the most viable threat to information systems today [2].

B. MALWARE

A computer is a tool that executes instructions or programs at an extremely fast pace. In general, when a computer executes a benign program, the program safely interacts with components of the computer in order to accomplish productive tasks. Malicious software executes instructions or programs that are not authorized by the user and can be embarrassing, frustrating, or cause damage to the system. According to Nash, malware, short for malicious software, is defined as:

> Programming (code, scripts, active content, and other software) designed to disrupt or deny operation, gather information that leads to loss of privacy or exploitation, gain unauthorized access to system resources, and other abusive behavior. Examples include various forms of

> adware, dialers, hijackware, slag code (logic bombs), spyware, Trojan horses, viruses, web bugs, and worms. [8]

The effects that malware has on a system can range from being a nuisance (adware, spyware, spam e-mails) to being devastating to the user or business, as in the case of virus logic bombs that delete critical files. A denial of service attack can be frustrating for users and can have even greater implications for businesses, such as loss of revenue. Root kits that allow an attacker unauthorized access to a user's or a business' data can result in some sort of loss or even identity theft, which poses a significant security risk. There are multitudes of types of malware and ways that malware can propagate throughout the Internet and networks. Additionally, the methods that are employed by hackers are constantly evolving. For the scope of this thesis, the term "malware" will pertain to worms, botnets, and viruses.

1. Worms

A worm is a self-replicating program that utilizes a computer network to send copies of itself to other computers and requires no human intervention to do so. There are three common characteristics of Internet worms according to Gu et al. [9]:

- Many worms generate a substantial volume of identical or similar traffic which provides the possibility of detecting them using their signatures and also the possibility of detecting them using network traffic analyzers such as Wireshark or intrusion detection systems such as Snort.

- They use random scanning to probe for vulnerable hosts which can also be detected by passive listening applications.
- Vulnerable hosts exhibit infection-like behavior when infected. That is, the host is first scanned, and then it sends out scans destined for the same port on other machines. This too can be detected by passive listening applications.

Worms typically have some sort of malicious payload or application that can be used to entice the user to visit a website, send data back to a central computer, create backdoors for further data extraction, or delete vital system files on the host computer. The first known Internet worm was the Morris worm in 1988, and other notable examples of worms are the Nimda worm and Code Red worm [1], [9].

2. Botnets

Botnets are networks of "bots," which are computers that have been infected by a worm and are subverted as remotely-directed hosts. Botnets are most commonly associated with Distributed Denial of Service (DDoS) attacks; however, according to Messmer,

> It's not just DDoS attacks that are associated with bots. Botnets are usually specialized, designed for criminal tasks that range from spam distribution; stealing identity credentials such as passwords, bank account data or credit cards

> and key-logging; click fraud; and warez (stealing intellectual property or obtaining pirated software). [10]

The main characteristic that distinguishes a bot from a virus or a worm is the command and control structure. Bots are typically designed with a command and control structure that allows for the subverted machines to be controlled by either a single server or a distributed command server. The command and control structures are generally coordinated over other protocols; for example, Hyper Text Transfer Protocol (HTTP) or Internet Relay Chat (IRC). This command structure lends itself to DDoS attacks since all of the bots could be simultaneously commanded to send traffic to a target server, which would overwhelm the target and could result in a loss of functionality, business, and revenue. Bots, like worms, will exhibit scanning behaviors as they try to expand the reach of the botnet and are, therefore, also detectable with traffic analysis tools such as Wireshark or Snort. Another characteristic of bots is that they can lay dormant for long periods of time until commanded by the control server to perform some function.

Some examples of botnets include Conficker, which is still active and is used to try to sell fake antivirus software; Gammima, which was used to steal gaming login information; and Zeus, which was used to steal banking information [1], [10].

3. Viruses

In his book, *The Art of Virus Research and Defense*, Peter Szor defines a computer virus as:

> Code that recursively replicates a possibly evolved copy of itself. Viruses infect a host file or system area, or they simply modify a reference to such objects to take control and then multiply again to form new generations. [11]

There are various ways that viruses are classified. One way that viruses are classified is by how they infect target hosts, such as boot records, files, and in-memory. They can also be classified by what computer architectures they target, such as processor types or operating systems; file systems and file formats targeted; or interpreted environments, such as scripts (PHP, Batch, Jscript, and Shell scripts) and macros. Viruses can also be classified by their defensive mechanisms, such as tunneling, retroviruses, armored, morphing, and encryption. Finally, viruses can be classified by the payload that they deliver to the target hosts, such as benign or harmless, destructive, data stealing, or denial of service [1], [10].

Viruses are typically combatted with the use of signature-based Intrusion Detection Systems (IDSs); which is a reactive approach once the virus has spread and caused some sort of damage. The success of IDSs depends on users or network administrators keeping their virus signature definitions up to date. Since viruses are code that resides somewhere on the infected host, the IDSs scan periodically based on the signature definitions to detect viruses. It is possible for viruses to morph as they spread making them more difficult to detect.

A well-known example of a computer virus is the "I Love You" virus. The I Love You Virus was a Visual Basic Script (VBS) LOVE-LETTER-FOR-YOU.TXT.VBS that was attached to an e-mail and tricked users into opening the attachment. Once the script was executed, it forwarded itself to all of the contacts in the victim's Microsoft Outlook contact list as well as overwriting numerous files on the victim's computer with malicious code. The I Love You Virus also created a number of registry keys so that it would be initialized when the infected machine booted. This virus exploited a Microsoft algorithm for hiding file extensions so that the extension appeared to be a benign ."TXT" file. The I Love You Virus spread across the world in less than a day, and in a week it is estimated that fifty million computers had been infected at an estimated cost of $5.5 billion [12], [13].

C. PROOF OF CONCEPT OF SOFTWARE TRAINING USING MALWARE MIMICS

In the thesis "Malware Mimics for Network Security Assessment" by CDR William Taff and LCDR Paul Salevski, the authors demonstrated that it was possible to create a software-based network training tool for network administrators and operators. They created a system based on the client-server relationship that allowed for modeling network traffic and behaviors of malware without any actual malware being introduced to the network. The authors showed that this tool could be used to provide training equivalent to that of red teams which could supplement the training provided by red teams.

The system that the authors created had the following characteristics:

- The system was designed to be safe for the network. If there was a loss of communication between the clients and the server it would be recognized as a termination of the exercise and the network would return to the normal operating state.

- The system only mimicked malware behaviors and no actual malware was ever introduced to the network. Using this model, we can mimic a multitude of malware for the training benefit of the users.
- The system constructed was distributed so that the trainer could be located anywhere on the network or even remotely to control the scenario.

Their thesis was a proof-of-concept for this training tool upon which we intend to expand.

D. HOW WE CURRENTLY TRAIN

1. Training Objective

In an effort to scope our discussion of training, we define the training objective as "the skill or behavior that we wish to reinforce." In this thesis, we will broaden the definition with respect to the complexity of the training objective. We will investigate some specific examples of malware/mal-behavior and the resulting trainee behaviors we wish to reinforce as a result of interaction with our software training tool [1].

2. Trainer-Trainee Relationship

The trainee is the person or group of people that we wish to train in accordance with a particular training objective. For the purpose of this thesis, the trainees will be network administrators and network operators that will gain a better understanding of their network through the use of our software based training system. The trainer is the person or entity that is administering the training to the trainee in order to evaluate their performance with respect to the training objective. Typically, the trainer for network administrators in the military is the red team that simulates the action of an adversary by utilizing the adversary's Tactics, Techniques, and Procedures (TTPs) in order to penetrate and exploit the network upon which we are conducting training. Other examples of the trainer for network operators are the network administrators or other more experienced operators providing training to the less experienced operators [1].

3. The Safety Observer

The safety observer is an important part of military training and, indeed, any training where risk is involved. The safety observer's responsibility is to oversee the trainer and the trainee to ensure that the training is conducted safely. The focus of the safety observer is not limited to any one specific aspect of the training scenario but is on how the training impacts the organization as a whole. There are instances when the complexity of the training is low enough that it does not warrant a third-party safety observer; for instance, when a network administrator is conducting training with a junior network operator on a single workstation. In this instance, the trainer can also fulfill the role of the safety observer. However, as the complexity of training scenarios increases and, in our case, the amount of critical mission functions that have been migrated to automated information systems grows, it becomes increasingly important that training scenarios are executed in a manner that does not bring unintended consequences upon the network or organization. This is when the safety observer is of critical importance. An example of a training scenario where safety observers play a critical role would be when a network is under attack on a ship that is in a close quarters battle scenario with other ships. If the attack on the network were to result in a loss of communications or radar equipment, the ship could be placed in danger while maneuvering in the vicinity of the other ships; which would place numerous personnel and assets at risk. The safety observer would be compelled to call a Training Time Out (TTO) in this case to restore control of systems to operators and allow the ships to maneuver to safety prior to recommencing training. During a training time out, the exercise is completely ceased and all parties involved in the training stop immediately, systems are restored and, in the case of maneuver of ships, there would be a predefined course for all ships to steer in order to avoid collision until the environment is deemed safe to recommence training. In this general example of a training scenario, we discussed the importance of the safety observer. In the next section, we will discuss how red teams are currently employed to conduct network training and some of the issues associated with this method of training.

4. Inherent Constraints Imposed by the Use of Red Teams

In the International Test and Evaluation Association Journal, David Aland provides pertinent and timely insight into the issues that are encountered with the way that we currently employ red teams for network training [7].

The first issue that the DoD is faced with when employing red teams to conduct network training is that red teams are a limited resource. With the growing number of mission critical functions that are being migrated to information systems, the agencies that sponsor red teams are experiencing an increasing demand for their services. Due to the fact that red teams are faced with budgetary constraints as well as a long lead time to develop and train skilled operators, an exercise planner simply cannot count on a red team being available for a particular exercise. Considering these constraints, red teams cannot be expected to expand the scope of the training they provide without having to cut back on the number of training assessments that they can conduct.

Another issue encountered with how we currently train with red teams is that the "customer" or the unit or organization sponsoring the training, typically imposes constraints or ground rules for the training so that the network training does not interfere with other training objectives that the units are facing. Commanders would be reluctant to expand the scope of network IA training without reassurance that this training would not interfere with other critical functions or training objectives. The complexity involved in training exercises, such as

COMPTUEX, and the fact that many events are interdependent and rely on information systems that would be subject to disruption if the networks were being attacked at the same time or if a red team were allowed to use their full arsenal of attacks result in artificialities in IA training or exercise events. These issues result in de facto limits on the training that red teams can conduct and negatively impact the quality of the training that is possible.

Standardization of results is another issue that is encountered when conducting network-training assessments with red teams. According to Aland,

> The traditional modus operandi of most red teams is to find and exploit a single vulnerability, making comparison of one event to another relatively difficult, with only a few common characteristics. [7]

This fact, coupled with the complexity of military networks, makes it very difficult to ascertain consistent feedback from red teams with respect to various training exercises since all of these variables lend themselves to unique training assessments of each unit that is observed.

The uniqueness of each training assessment conducted by red teams makes it difficult if not impossible to determine trends or common problems that are affecting the security of DoD networks as a whole. There are significant advantages to be gained by having a core set of training events that are conducted against all DoD networks that are assessed by red teams. The results of assessments could achieve a greater level of standardization between assessed entities, which would allow for creation of a database that could be used to compare results from subsequent assessments and determine trends. These trends could then be analyzed to statistically determine the rate of success and failure for particular attacks, as well as to identify root causes of DoD network vulnerabilities as a whole.

These constraints give way to the information system solution upon which we are continuing development that will allow red teams to expand the scope of their training assessments without requiring additional time, personnel or other resources.

E. EXAMPLE TRAINING SCENARIO

In the planning period leading up to the exercise or Pre-exercise (PRE-EX) phase, the agency responsible for conducting the training would develop a tailored training scenario to achieve the specified training objectives. The following describes such an activity.

For this example, the training objective will be to identify and take the appropriate steps to combat a worm propagating on each ship’s network. The trainee’s for this exercise will be the network administrators and operators of a Carrier Strike Group operating in the Virginia Capes as part of their pre-deployment training cycle. The network training will be conducted simultaneously with other training events that the Strike Group is conducting; such as flight operations on the Aircraft Carrier, tactical maneuvering of the ships in the Strike Group, and an Anti-Submarine Warfare exercise.

Prior to commencement of the exercise (COMEX), the ship’s Combat Systems Training Teams (CSTT) would receive the PRE-EX directive, which outlines the training objectives as well as amplifying information on the scenario, such as behaviors which will be exhibited by the networks, Training Time Out procedures, etc. The individual ship's CSTTs will serve as the notional "white cells" and act as the safety observers for the exercise. Each CSTT will conduct an exercise brief to establish roles and responsibilities and review safety procedures, as well as expected actions to be taken by the trainees. Once all of the safety observers are in place, the ship would be placed in a Combat Systems Training Team environment and the status would be communicated accordingly throughout the ship.

Upon commencement of the exercise (COMEX), the entity conducting training on the network (red team), physically located at Fort Meade, MD, will issue the command from their scenario generation server to the trainee ship's scenario execution servers to execute the appropriate modules to exhibit the behaviors and signatures of a worm propagating on the network. Once the local exercise server receives this command message it will issue a directive to a predetermined number of hosts on the network to begin exhibiting a behavior (in this instance, port scans). Once these hosts begin scanning other potentially vulnerable hosts, the server will command additional hosts to begin conducting port scans in order to simulate the spread of the worm on the network. The only effect that our software will have on the network will be an increase in benign traffic traversing the network. The increase in traffic, as well as other alerts to the network operators, will flag the presence of something out of the ordinary on the network and will elicit them to investigate further to determine the cause of the errant behavior. Upon further investigation the network administrators and operators should identify the behavior as coming from a worm propagating on their network and should take appropriate measures to quarantine the affected machines as well as stop the propagation. Finally, the administrators should report the infection to the higher echelon in the Chain of Command, which would be relayed to the red team and the exercise would be halted.

After the exercise, the results in the database could be compared with previous exercise results to determine trends such as success/failure rate of identifying the malicious behavior, time to identify, quarantine of affected hosts.

F. SUMMARY

In this chapter, we have discussed how red teams are utilized to provide realistic training. We have also discussed some instances of malicious software and the effects that they can have on networks and computers. Additionally, we have discussed how some of the threats are employed by red teams and their ethical hackers in order to train network administrators and operators. We have also discussed how a software-based training tool has been proven viable. Finally, we have discussed how we currently train network administrators and operators and asserted some of the constraints that are inherent with the use of red teams. In the following chapter, we will assert how we can augment the training provided by red teams by expanding the software based training tool. Additionally, we will explore the desired behaviors that our system will exhibit.

III. DESIGN CONSIDERATIONS AND TEST PLATFORM

In this chapter, we will discuss the current state of MAST. We will also briefly discuss the benefits of the MAST system for enhancing DoD network security. Additionally, we will discuss the implementation platform hardware and software that we are utilizing for further development. Finally, we will introduce and discuss in depth the Host Based Security System (HBSS).

A. OVERVIEW OF MALICIOUS ACTIVITY SIMULATION TOOL (MAST)

The MAST system implements a variety of new features and improvements over the previous version of software [1]. The MAST software utilizes the client-server architecture and allows simulated adversaries (red teams) and trusted agents (blue teams) to leverage their existing skill sets and conduct training without an increase in risk while operating within the prescribed limits. This training tool provides trainers with a whole new set of tools to test the trainees' reaction to particular malware, by which to evaluate them with respect to a given training objective. The behaviors that our software mimics also allows for the same attacks and behaviors to be conducted on various DoD networks in a similar manner to allow for consistent assessments and better, more-consistent feedback from training assessments.

1. System Design

The system is designed so that the behaviors and signatures of any particular malware are externally observable and elicit appropriate responses from the network operators and administrators. The software consists of a remote central server that commands a server local to the trainee network to execute a training scenario. The local server then commands the malware mimic clients running on the workstations of the trainee network to exhibit specific malicious behaviors; for example, conducting port scans of other hosts on the network to mimic the behavior of a worm propagating on the network. The overall system architecture is depicted in Figure 1. The software poses no actual risk to the network since it only mimics malware behaviors and does not infect any host with actual malware. The result is externally observable malicious behavior without actually introducing any malicious code on the network.

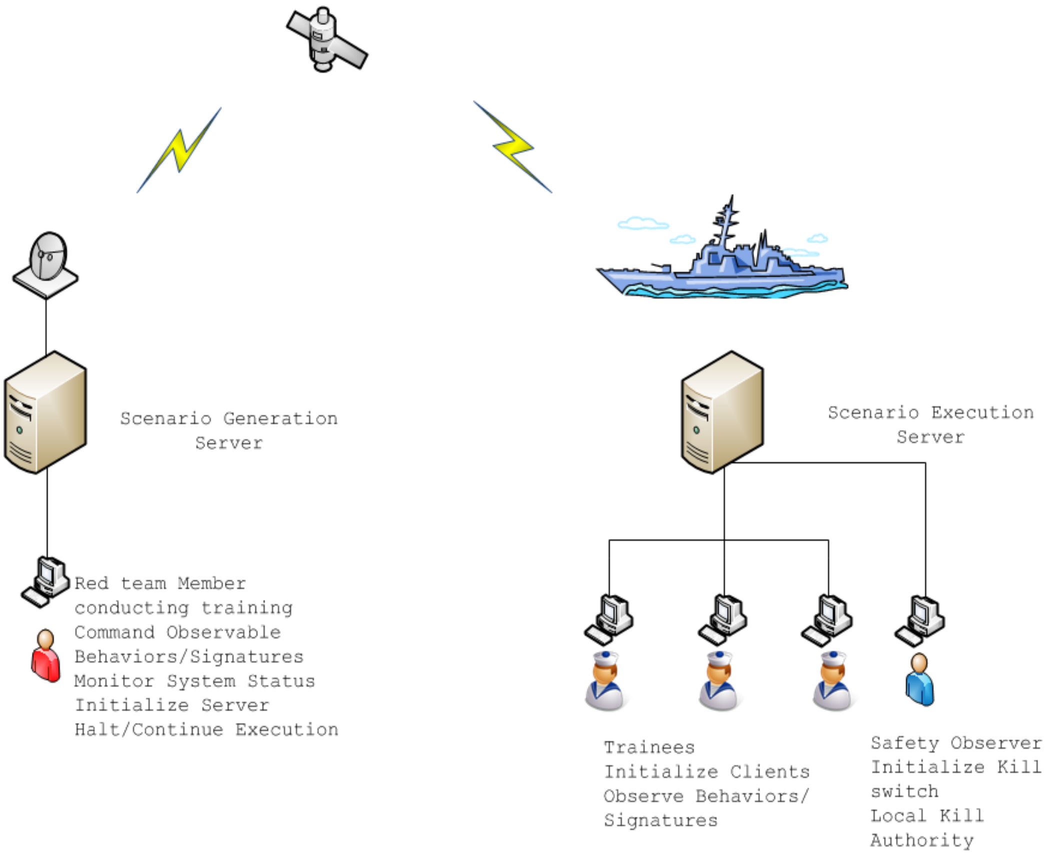
Scenario Generation Server
Scenario Execution Server
Red team Member conducting training
Command Observable Behaviors/Signatures
Monitor System Status
Initialize Server
Halt/Continue Execution
Trainees
Initialize Clients
Observe Behaviors/ Signatures
Safety Observer
Initialize Kill switch
Local Kill Authority

Figure 1. The MAST Architecture

2. Server Design

The scenario generation server shown in Figure 1 is remotely collocated with the entity conducting the training, such as the NSA red team in Fort Meade, Maryland. This scenario generation server is the central hub from which training can be conducted on various units or organizations remotely. In order to conduct training with a particular entity, such as a ship or a strike group, the scenario generation server establishes a logical connection to the ship’s or multiple ships’ scenario execution servers via the Global Information Grid (GIG). Once connection is established and all parties are ready to commence training, the red team begins sending commands via the scenario generation server to the ship’s scenario execution server. Upon receipt of the command to emulate a certain malicious behavior, the scenario execution server verifies that the required modules to execute the commanded behavior are installed. Once the modules are verified as installed, the scenario execution server commands a predetermined number of hosts to begin exhibiting the malicious behavior.

In addition to the ability to conduct training remotely, the trainers are able to monitor training and receive feedback upon completion of the training scenario. The scenario generation server also has the capability to pause or halt the training scenario remotely. The scenario execution server is also able to pause or halt training locally if the operational conditions warrant such an action. Additionally, the scenario execution servers (again, local to each network) has the capability to conduct local training assessments throughout the training cycle without the need for a red team to conduct individual unit training. This feature is similar to the Navigation Seamanship Shiphandling Trainer (NSST) that is currently in use on ships throughout the Navy to conduct “in-house” ship handling training without the need to get underway, that is, to leave the pier. NSST has allowed junior officers to increase their proficiency while saving the Navy substantial amounts of money associated with getting ships underway. Similarly, our training tool is designed to provide network administrators with the capability of conducting training locally, throughout the training cycle, without an increased demand for red teams. The value added from this feature will dramatically increase the ability of network administrators and operators to identify malicious behavior and defend their networks while decreasing the training costs associated with utilizing red teams. An additional benefit in MAST to DoD network administrators and operators is that they do not have to wait to be assessed by a red team in order to get experience in identifying and combating malicious software. This functionality to allow for local training is also a new feature of the MAST system. Finally, the scenario generation server and the scenario execution server have databases that log the results of a particular training scenario for comparison with past/future training assessments. These databases are a new feature which allow for better, more consistent feedback on training scenarios than what our current training methods are capable of producing.

3. Host Design

The hosts on the trainee network have a lightweight software package, the malware mimic client, which when commanded begins exhibiting the desired malicious behavior. The malware mimic clients are logically connected to the scenario execution server. When the network is not in a training environment, the malware mimic clients continue to run idle, in an effectively dormant state. However, once commanded to exhibit a malicious behavior by the scenario execution server, they verify that the necessary modules are installed and then commence exhibiting the desired behaviors. The trainee's interaction with the system is to observe the malicious behaviors and react to them appropriately.

4. Safety Features

Our software includes various safety measures to ensure that the scope of network training can be expanded without a concomitant increase in risk associated with the training.

Prior to commencing training on an entity from a remote location, the scenario execution server on the trainee's network would have to be placed in training mode. This feature prevents remote training outside of a prescribed training event from occurring without the ship's permission. As depicted in Figure 1, a local software-based "kill switch" is also implemented on the execution server so that if local conditions warrant that the training be ceased immediately (i.e., A Training Time Out), the safety observer or network administrator on the trainee ship can stop the scenario immediately without the delay of notifying the remote trainer. When the “kill switch” is activated, the scenario execution server immediately commands all of the malware mimic clients that are currently exhibiting malicious behavior to halt. The malware mimic clients immediately rollback to the idle state that they were in prior to the training exercise and the network, as a result of this rollback action, returns to normal operation. These features are analogous to the “two-key” safety systems in place with ballistic missiles or other weapons systems.

For ships operating at sea, there is the possibility of interruption of network connectivity between the scenario generation server and the scenario execution server. The “kill switch” also solves this problem, since the scenario can be allowed to execute on the trainee’s ship and can be stopped when training is complete. In this case, the feedback generated from the training scenario is stored locally on the ship and transmitted back to the entity conducting the training once connectivity is restored.

An additional safety feature of the system deals with a loss of connectivity between the scenario execution server and the hosts on the network. When a scenario is in progress and the malware mimic clients on the hosts exhibiting the malicious behavior lose contact with the scenario execution server, they immediately cease the behavior that they are exhibiting and rollback to the idle state. This feature prevents the malware mimic clients from continuing “headless.” That is, they will not continue to exhibit their malicious behavior independent of the scenario execution server. The aforementioned safety features are improvements to the original safety features of the previous system.

5. System Overhead

The malware mimic client software component of MAST is expected to be installed on hosts on DoD networks as part of a base installation, along with the local scenario execution server for each network.

As new modules are created to mimic the most recent threats that our networks are facing, they would be "pushed" out to all pertinent network administrators in a manner similar to how software patches or Information Assurance Vulnerability Alerts (IAVAs) are pushed to the commands. With this system to distribute the latest modules in place, it should be possible for the trainee command to install and maintain the most recent software and modules prior to conducting a training assessment. However, if a trainee command's scenario execution server receives a command to execute a module that is not installed from the trainer's scenario generation server, an error message would be returned to the trainer. Upon receipt of an error message, the trainer will push the latest modules to the trainee's scenario execution server, which, in turn will distribute the module software update to the hosts that are active on the trainee's network. In practice, however, network administrators would be required to ensure that their network has the most recent software modules in place in preparation for an upcoming training exercise. This feature is new in the MAST system and is designed to reduce the burden on network administrators while ensuring that the system is ready to conduct training with the latest malware modules.

Each command’s scenario execution server will “know” the current status of its network (i.e., which hosts are online) and will choose the hosts to begin executing the particular module from these active hosts. In the event that the scenario execution server local to the trainee’s network commands a host to execute a particular module and that host does not have the module installed (possibly due to the host being reimaged), the scenario execution server will push the latest module software to the host or choose another host to execute the module. This feature enables the training to continue on the network without delay if an individual host is not updated.

6. Benefits of MAST System

The benefits of MAST are two-fold; we can leverage the red team’s current skill set, particularly in developing training or assessment scenarios, while establishing a core set of training events that can be repeated that allow for more consistent feedback from training scenarios. Additionally, the value of this training tool is not only during red team assessments, but the individual units are able to conduct local, in-house training throughout the training cycle to better prepare the network operators and administrators to deal with real-world threats. Finally, the safety features implemented by the MAST system allow for more frequent and thorough training while keeping risk well within operational limits.

B. IMPLEMENTATION PLATFORM

In an effort to ensure that our system provides the most realistic training possible, we have designed our proof-of-concept implementation platform to simulate a mock shipboard network. By using the software that is currently in use on Navy ships, we aim to prove that the MAST system is a viable training tool for system operators and administrators throughout the DoD.

1. Hardware

The hardware that we are using to implement the MAST system is designed to support virtualization of a mock Cruiser (CG-71) shipboard network. We are using three Dell PowerEdge R610 servers to run VMware server management software and the associated virtual machines. The hardware specifications for the Dell servers are as follows:

- Server 1: 2TB Hard Drive, 32GB RAM, (2)Intel® Xeon® Quad-core 2.4GHz processors
- Server 2: 1TB hard drive, 16GB RAM, (2)Intel® Xeon® Quad-core 2.4GHz processors
- Server 3: 1TB hard drive, 16GB RAM, (2)Intel® Xeon® Quad-core 2.4GHz processors

The Dell servers are designed to enhance virtualization capabilities and provide sufficient physical memory to support multiple Virtual Machines (VMs). All three servers are connected using a Dell 2716 Gigabit switch making the network fully switched since each segment is only connecting each respective server and the switch. This configuration allows for full duplex communication between the servers and the switch. Thus, the packets can travel from server to switch and from switch to server simultaneously thereby minimizing latency on our test network. Each server is connected with two Ethernet cables which are "trunked," meaning that the two 1GB capacity Ethernet cables are seen as a single 2GB "pipe" to increase speed of file transfers between the servers. This physical configuration enabled us to make the most efficient use of our physical resources thereby enabling us to implement an accurate model of a shipboard network through virtualization as depicted in Figure 2. Additionally, we used a Cisco 2811 router that serves as the access point for remote hosts to connect to the VMs. Finally, we are using a Dell 1920 Uninterruptable Power Supply (UPS) to ensure that we have time to safely shut down our system in the event of a loss of power.

2. Software

We used VMware products to virtualize our implementation platform to replicate a mock shipboard network. By employing virtualization, we were able to simulate an entire network without having to use physical hosts, thereby reducing the amount of space and hardware necessary as well as eliminating the need for cable runs, etc. According to VMware, a virtual machine is "a tightly isolated software container that can run its own operating systems and applications as if it were a physical computer" [14]. A VM behaves exactly like a physical computer and has software based CPU, RAM, hard drive, and network interface cards (NIC). The user perceives the VM as a physical computer when in reality it is just another program running on the host computer.

We used VMware vSphere to implement virtualization on our servers; it has two components: VMware ESXI version 5.0 and vCenter Server. ESXI is the hypervisor and is installed on the "bare metal," that is, it does not require a host operating system to run. The VMs are completely encapsulated and the hypervisor handles all calls for host resources, such as the processor, and device drivers, by each VM. The hypervisor coordinates and schedules all resource requests from all of the VMs to the host upon which they are running. The management of the VMs is handled by vCenter Server, which allows us to centralize the management, configuration, and monitoring of the VMs. To access vCenter Server, we used vCenter vSphere Client, which allowed us to add VMs to the servers and coordinate the actions of all such VMs on the servers as depicted in Figure 2. The VMs that we used were already configured to work with the hypervisor. However, if they were not previously configured to work with the hypervisor, we would have had to use the VMware converter which takes different types of VMs and makes them compatible with the hypervisor.

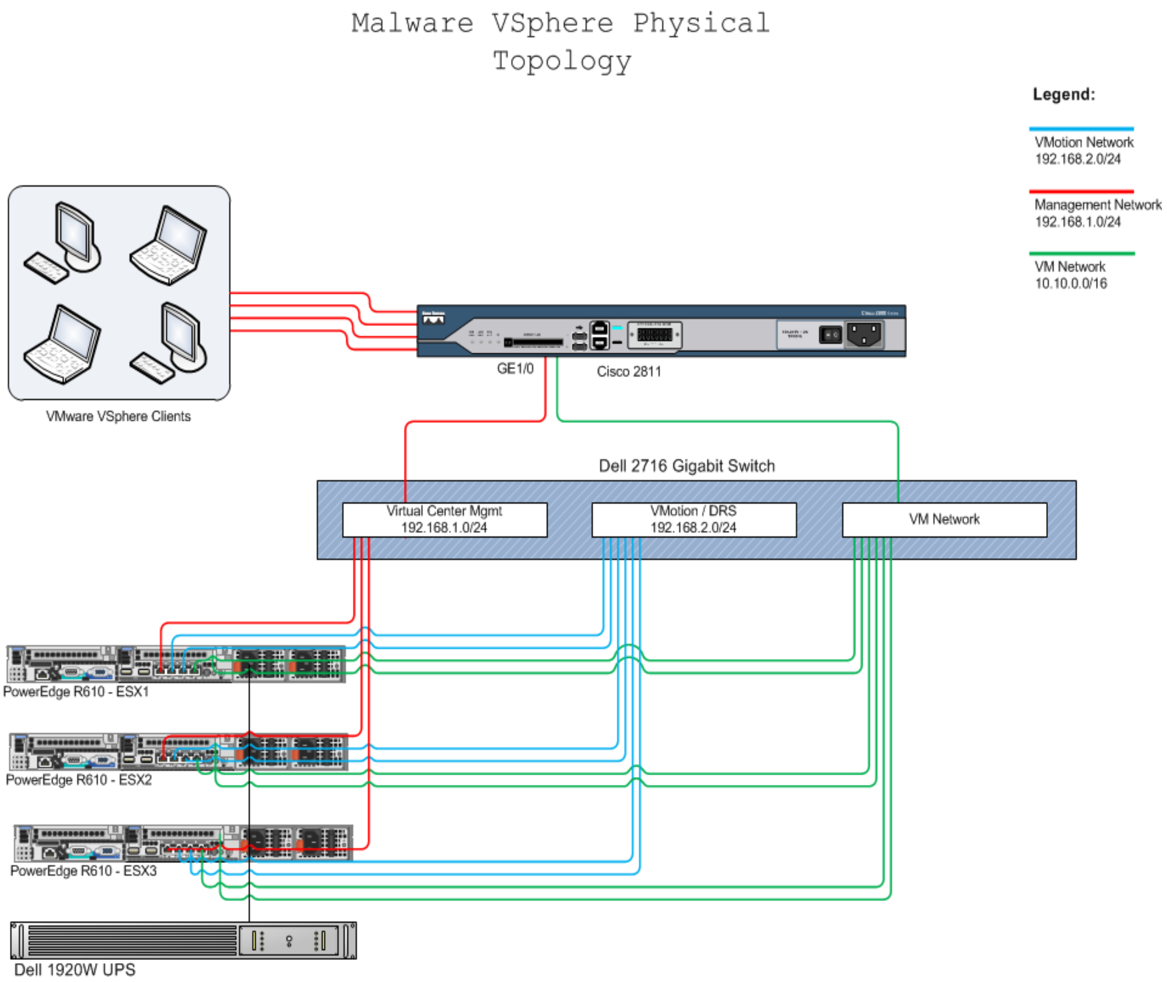

Figure 2. Physical Topology of Implementation Platform.

We organized our servers into a cluster to maximize the efficiency of the server hardware. A cluster is a group of hosts that share resources and a management interface and it effectively makes the three servers appear as one resource to VMware. An advantage of configuring our servers in this manner is that it allowed us to use VMotion Dynamic Resource Scheduling (DRS), which allows for dynamically balancing the VM load across the three servers as well as manual load balancing [15]. However, to use the dynamic VM balancing, we would need a Storage Area Network (SAN), which is a separate network of block level storage that appears to the servers as locally attached storage. Consequently, if a particular host’s resources get bogged down with its current VM workload, we have the capability to migrate running VMs to another host with lower resource utilization to ensure that all VMs have sufficient resources, thereby minimizing simulation-induced latency on the network.

Additionally, we obtained a commercial license for VMware vCenter that allows us to run an unlimited number of VMs on our servers (subject to physical memory and processor constraints). This allows us to overcome a licensing constraint that the previous Malware Mimic project faced, which only allowed for ten VMs to run on each server.

3. COMPOSE CG-71 Virtual Machines

To support further development of the MAST system and to ensure that our software provides realistic training on current DoD networks, we used the Common PC Operating System Environment (COMPOSE) CG-71 VMs (ISNS AN/USQ-153(V)9), which are VM representations of the actual servers and workstations that make up the unclassified enclave on CG-71. These VMs allow us to virtualize the exact configuration of a shipboard network with which operational network administrators in the fleet currently work.

We obtained nine VMs from SPAWARSYSCEN Pacific contractor ManTech, San Diego, CA, to simulate a shipboard network on our implementation platform.

a. Integrated Shipboard Network System (ISNS) Domain Controllers (1,2)

Microsoft Windows Server 2003 Standard Edition that provides the following services: COMPOSE data server, primary/secondary DHCP (Dynamic Host Configuration Protocol), primary/secondary DNS (Domain Name System) with Active Directory integrated, Symantec antivirus server, and other associated services (file/print services, etc.).

b. ISNS Exchange Server

Microsoft Windows Server 2003 Standard Edition, Exchange Server Standard Edition, Internet Information Server (IIS) for Simple Mail Transfer Protocol (SMTP), Network News Transfer Protocol (NNTP), Web Services (HTTP/HTTPS/FTP).

c. ISNS System Management Server

Microsoft Windows Server 2003 Standard Edition, SQL Server 2005 Standard Edition, Internet Information Server (IIS) for Simple Mail Transfer Protocol (SMTP), Network News Transfer Protocol (NNTP), Web Services (HTTP/HTTPS/FTP). Also provides SMS distribution point and server.

d. CND-OSE HBSS Server

Computer Network Defense-Operating System Environment: Microsoft Windows Server 2003 Standard Edition, Host Based Security System server that includes the ePolicy Orchestrator (ePO).

e. CND-OSE MSSQL Server

Microsoft Windows Server 2003 Standard Edition, Microsoft Structured Query Language server and database for HBSS and Secure Configuration Compliance Validation Initiative (SCCVI).

f. CG-71 COMPOSE Server

Microsoft Windows Server 2003 (32 bit), Common PC Operating System Environment server.

g. CG-71 COMPOSE SCCVI

Microsoft Windows XP Professional (32 bit), Secure Configuration Compliance Validation Initiative client that works with HBSS to ensure compliance of workstations.

h. CG-71 COMPOSE Workstations

Microsoft Windows XP Professional (32 bit), McAfee Agent running which interacts with and reports to HBSS.

By using VMs of the unclassified enclave, we leverage the power of virtualization while creating the most realistic implementation platform for further development of the MAST system. The COMPOSE/HBSS servers require roughly 10GB RAM and with our current hardware configuration (64GB total RAM), we were able to run 25-30 Workstation VMs (1.5GB RAM each) before performance began to degrade due to resource limitations. These limitations will not impede further development of the MAST system; however, they will need to be resolved before we can test the scalability of the system.

C. HBSS

Host Based Security System (HBSS) is being deployed by the DoD to provide security for Windows and Unix servers and workstations. A thorough discussion of HBSS is necessary since the MAST system has to interact with HBSS in order to function properly and provide realistic training to DoD network system administrators and operators.

HBSS provides host-based security to the network through behavioral, signature, desktop-firewall, and application-blocking protections. Behavioral rules are established to identify a profile of network activity; deviation from these profiles will result in a system alert. Signature-based, malicious-activity protection is provided by the Host Intrusion Prevention System (HIPS). HIPS agents cross-check host activities against the malicious-activity signature database to determine whether or not activity is malicious. If an activity is determined to be malicious, an alert is sent to the McAfee ePolicy Orchestrator console, which is described below. HBSS provides desktop, i.e., individual host platform, firewall protection by establishing a filter between the host systems and the network or Internet. All network traffic to and from each host is scanned at the packet level and compared with the list of firewall rules. Finally, the application-blocking capability prevents users from launching certain executable files on the host systems. HBSS provides network administrators and operators with tools to prevent, detect, track, and remedy malicious computer activities and incidents across all DoD networks [16].

HBSS, as it is currently deployed in MAST, is virtualized using VMware ESXI to host the components of HBSS on a single server. The software components that are virtualized are: Microsoft SQL Server Management Studio (provides user database for HBSS), Secure Configuration Compliance Validation Initiative (SCCVI), and HBSS. We have essentially virtualized the rest of the CG-71 network for our implementation platform.

1. McAfee ePolicy Orchestrator (ePO)

HBSS behavior is governed by policies and the ePolicy Orchestrator (ePO) Server is the central policy management point for all of the systems that HBSS manages. The ePO delivers security policies and tasks, controls policy updates, and processes alarms (events) for all HBSS managed hosts. The management of the various security products (HIPS, Rogue System Detection (RSD), etc.) is accomplished through the combination of product policies and client tasks. Product policies ensure that a product’s features are configured and perform correctly. Client tasks are the scheduled actions that run on the managed systems hosting the client side software [16].

The MAST system has to be added as an exception to the ePO in order to allow it to run and not be blocked by the Host Intrusion Prevention System (HIPS).

2. McAfee Agent

The McAfee Agent is the distributed client-side software that securely communicates information and enforcement of policies for each host and the ePO. For each managed host on the network, the agent retrieves updates, executes scheduled tasks, enforces policies, and reports malicious activity events to the ePO server. The Agent-to-Server Communication Interval (ASCI) determines how frequently the agent contacts the agent-handler in the ePO server for policy updates. The default ASCI is 30 minutes for small deck ships (Destroyers, Frigates, etc.) and 60 minutes for large deck ships (Aircraft Carriers, Amphibious ships). If a policy update is urgent and needs to be sent to all clients immediately to address a threat to the network, a "wake-up" call can be sent to all agents on managed systems to force them to receive the update immediately. However, this will slow down the network temporarily due to the increase in network traffic to all hosts [16].

The McAfee Agent also encompasses the following product agents:

- Host Intrusion Prevention System (HIPS)
- Device Control Module (DCM) Plug-in
- Asset Baseline Monitor (ABM) Plug-in
- Policy Auditor (PA) Plug-in

3. McAfee Host Intrusion Prevention System (HIPS)

HIPS provides several fundamental security features, including application blocking and firewalls that, when combined, reduce risk for managed hosts. HIPS utilizes signatures, behavioral-based rules, and host-based firewalls to prevent attacks. HIPS is comprised of three separate features: the Intrusion Prevention System (IPS), the firewall, and application blocking [16]. Each is described below.

a. Intrusion Prevention System (IPS)

The IPS feature monitors all system and Application Program interface (API) calls and blocks program calls that are determined to be malicious in nature, based on signatures. The IPS monitors individual host's program calls, as well as network program calls. IPS uses a database of signature rules that is installed with each McAfee Agent and updated as new attacks are discovered. IPS signatures are categorized by severity of the threat (high, medium, low, information) and set the actions to be taken by the IPS when a particular signature is matched. The actions taken are configurable by system administrators and range from ePO malicious-event notification and system logging to completely blocking the application. The default configuration of the IPS will automatically block the host on which malicious activity is detected from the network for ten minutes, essentially denying service to that machine in an attempt to isolate the malicious activity. With a policy exception for the MAST system in the ePO, the MAST system will be able to generate simulated malicious activity for the system administrators to detect and to appropriately respond [16].

b. HIPS Firewall

The HIPS firewall protects managed hosts from intrusions that compromise data, applications, or the host operating system. The firewall protects hosts by analyzing network traffic at different layers of the Open System Interconnection (OSI) networking protocol model, based on specific protocol criteria for each layer, to restrict processing of potentially malicious network traffic. The firewall is a “Stateful” firewall, meaning that it keeps track of the state of network connections and traffic traversing the network. Stateful packet filtering is accomplished at the Transport Layer (Layer 4 of the OSI model) by examining TCP/UDP/ICMP traffic headers and comparing the packets against existing firewall rules as depicted in Figure 3. If the packet matches a firewall “allow” rule, the packet is forwarded and added to the state table. The state table dynamically tracks network connections previously matched against the static firewall rule set for TCP/ICMP, and therefore reflects the current state of the TCP/ICMP protocols on the network. Additionally, stateful packet inspection is done at Layer 7 of the OSI network stack model and is the process of inspecting actual application data in packets and tracking of commands sent to and from applications as depicted in Figure 4. This combination of stateful packet filtering and inspection provides a strong representation of the host’s current connection state.

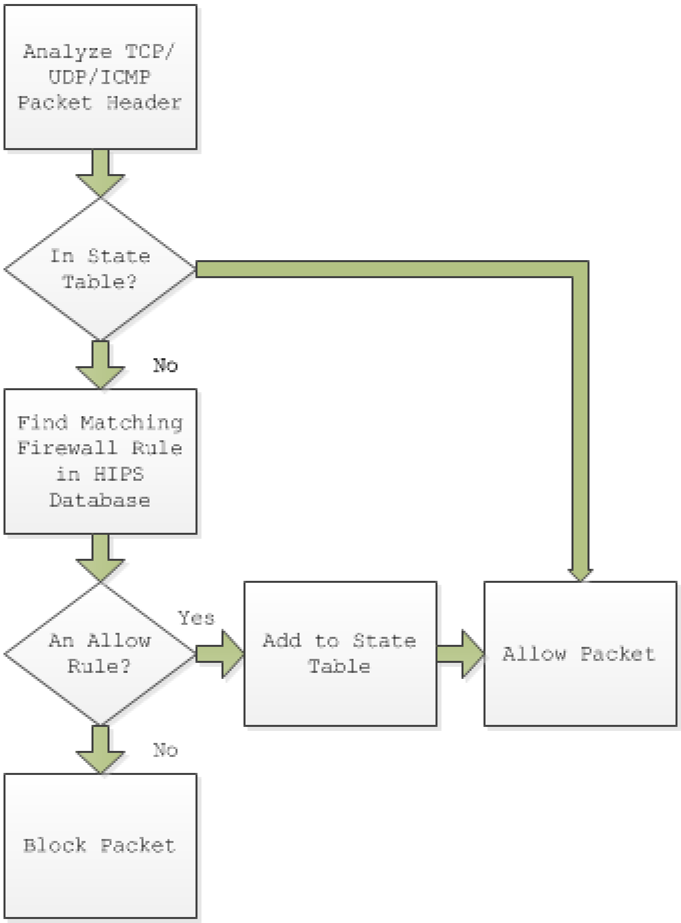
Analyze TCP/ UDP/ICMP Packet Header
In State Table?
No
Find Matching Firewall Rule in HIPS Database
An Allow Rule?
Yes
Add to State Table
Allow Packet
No
Block Packet

Figure 3. Firewall Stateful Filtering.

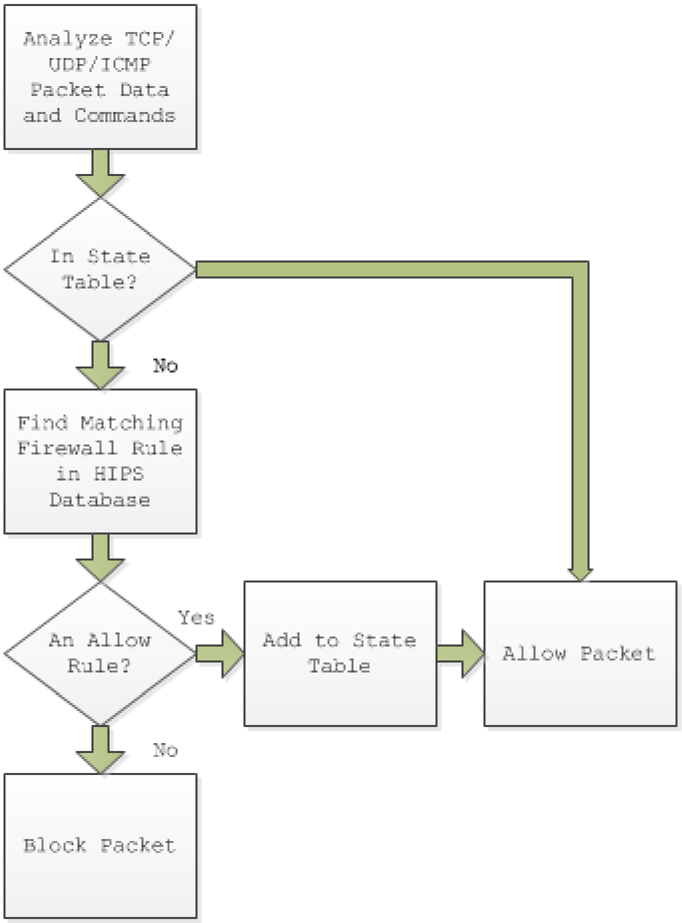
Analyze TCP/ UDP/ICMP Packet Data and Commands
In State Table?
No
Find Matching Firewall Rule in HIPS Database
An Allow Rule?
Yes
Add to State Table
Allow Packet
No
Block Packet

Figure 4. Firewall Stateful Inspection.

The firewall rules are used to determine how to handle network traffic. HIPS uses precedence of its firewall rules to determine if traffic should be allowed or blocked, that is, the traffic is compared to the first rule in the firewall rule list and if it matches, the traffic is forwarded and an entry made in the state table. If the traffic does not match the first rule, it is compared to the next rule in the firewall list, so on and so forth. If the traffic is not allowed by the last rule in the firewall list, the traffic is blocked, that is, discarded without further processing [16].

c. HIPS Application Blocking

HIPS Application Blocking provides the capability to block application “creation” and application “hooking.” Application creation monitors applications that are trying to execute. Application rules are similar to firewall rules and allow specific applications to launch. Application hooking monitors applications that are trying to bind or “hook” themselves to other applications. While application hooking is an essential part of modern operating systems, it can be used nefariously to run malware. A legitimate example of application hooking is an application that intercepts a keyboard or mouse “event” message to modify the functionality of the application. Programs or applications that are allowed to hook to other applications are also checked against the application rule set [16]. The MAST system has to be added to the application rule list in the ePO to be allowed to run properly.

4. Device Control Module (DCM)

The DCM restricts access to peripheral devices such as thumb drives and other removable storage to prevent unauthorized data extraction. McAfee device control prevents unauthorized use of removable media using content aware data protection. The DCM can be used to limit what data can be written to removable media, or to block access to removable media entirely [16].

5. McAfee Asset Baseline Module (ABM)

The ABM is an extension of the ePO and provides system and file activity monitoring. The ABM provides automated support for Information Operations Condition (INFOCON) procedures. INFOCON provides a system framework by which commanders “can increase the measurable readiness of their networks to match operational priorities” [17]. INFOCON levels provide alert readiness-postures similar to Defense Readiness Condition (DEFCON) for the Armed Forces. The INFOCON levels range from “5,” where there are no apparent attacks against DoD information systems, to “1,” where DoD information systems are currently under attack and are configured for the maximum defensive readiness-posture.

The ABM also provides system-level monitoring, such as changes to Windows registry-keys, services, ports, files, and local/user groups. The ABM provides the capability to conduct baseline scans and activity scans. Baseline scans record the state of a system’s monitored activities at a particular point in time. Activity scans record any changes to the system’s monitored activities since the last baseline scan [16]. After the MAST system is installed, it will be necessary to conduct a new baseline scan to capture the state of the system’s monitored activities with the malware mimic clients running.

6. McAfee Policy Auditor (PA)

The PA provides the ability to validate the integrity of a system by scanning for configuration settings and options. The PA automates the processes required to conduct internal and external IT policy audits. By ensuring that host systems are configured correctly, the PA provides an overall check of system health [16].

7. McAfee Virus Scan Enterprise (VSE)

The VSE offers easily scalable protection, fast performance, and mobile design to protect your environment from viruses, worms and, Trojan horses. This product is not currently in use by COMPOSE; however, once this product is implemented, it will be necessary to include a policy exception to allow the MAST system to run its simulated malicious activity [16].

8. McAfee Rogue System Detection (RSD)

The RSD provides real-time discovery of “rogue” systems, that is, systems that have a network interface card (NIC) connected to the network and do not have a McAfee Agent installed from the network’s ePO. The RSD utilizes passive rogue system sensors placed throughout the network that listen to network broadcasts and Dynamic Host Configuration Protocol (DHCP) traffic to detect systems connected to the network [16]. The RSD sensors detect rogue systems when they send a broadcast “ARP” request on a VLAN segment, and if the sensor does not have an entry for the system that sent the ARP, it will report this system to the ePolicy Orchestrator. The ePO then checks to see if the host that sent the ARP request has a McAfee Agent installed and, if no agent is installed, the system is labeled as rogue. Similarly, if a host attached to the network sends a DHCP request for an IP address from the DHCP server, it will be reported to the ePO server and checked to see if it has a McAfee Agent installed and labeled accordingly.

D. SUMMARY

In this chapter, we discussed the design considerations associated with the MAST system. Additionally, we discussed the implementation platform hardware and software to create a realistic development platform for the MAST system. Finally, we discussed the Host Based Security System with which our system will interact to provide realistic training to DoD network administrators and operators. In the next chapter we will perform a verification and validation of the MAST system concept and assert how the MAST system can simulate many of red team and “ethical hacker” methods to reduce the burden currently placed on DoD red teams. We will also explore the cost effectiveness of the MAST system concept.

IV. CRITICAL ANALYSIS OF RED TEAMS VS. MAST

In this chapter, we discuss methodologies that are employed by ethical hackers and red teams to conduct a security assessment or penetration test of a given network as well as the advantages and disadvantages of their approaches. We discuss the five phases which red teams and ethical hackers use to advance an attack. We will then assert how the MAST system can achieve a high level of effectiveness of training and improve upon the training that red teams currently provide. Additionally, we will conduct a verification and validation analysis of the suitability of the Malware Mimic concept as a methodology for conducting network administrator network security training and awareness and develop a strategy by which the effectiveness of the MAST system for increasing such network security awareness and elevating the information assurance posture of distributed command networks can be measured.

A. RECONNAISSANCE

As previously discussed in Chapter II, ethical hackers and red teams typically utilize a five phased approach to advance an attack against a target network: reconnaissance, scanning, gaining access, maintaining access, and covering their tracks. In this and the following sections we discuss some of their methods and how the MAST system will enhance their capabilities.

1. Red Team Methods

Red teams conduct reconnaissance on a target network by utilizing a technique known as "footprinting." According to the Certified Ethical Hacker Manual, "Footprinting refers to uncovering and collecting as much information as possible about a target network" [6]. Red teams utilize footprinting to gather valuable system level information about target networks such as operating systems and other software version information as well as individual account details, server names, and database schema. DoD red teams may not necessarily have to rely on the various methods of footprinting to gather information about target networks due to the fact that they have a lot of this information from their knowledge of systems deployed on DoD networks. However, in some cases, red teams use some of the following footprinting methods to gather further information about the configuration of the target network.

a. Internet Footprinting

Internet footprinting consists of extracting data about a target network using search engines and other freely available tools on the Internet. An example of how a red team could use information that is easily found on the Internet to gain further information about a target network would be to visit the Command's official website to gather information about the Command's leaders, i.e., Commanding Officer, Executive Officer, and Command Master Chief, such as e-mail address and other information. This information could then be used to mount social engineering phishing attacks against crew members. For example, a red team could spoof e-mails from the leadership of the target command to junior personnel in the command to determine the effectiveness of the command's training program by attempting to coax personnel to divulge sensitive information, such as information about the command's schedule. Likewise, the red team could spoof the e-mail address of the network administrator to see if any personnel would divulge information about their accounts, such as their password for example, which the red team could use to attempt to gain further access to the target network.

Another tool that red teams utilize to discover information about a target network is the "whois" query. Whois is a utility that is available in various Unix/Linux distributions and there are tools that implement similar functionality readily available online. Whois queries return information about domain owners from Regional Internet Registries (RIRs) such as domain name details, domain name servers, NetRange (IP address range of the target domain), administrative contact information, phone numbers, etc. With this information, red teams can gain further information about the target network using DNS footprinting.

b. DNS Footprinting

DNS footprinting enables a red team to extract further information about the target network through various DNS interrogation tools. DNS footprinting tools query the target network's DNS records to gather information about the topology of the network. Some examples of DNS records and the information that they reveal are: the A record which points to target host’s IP address, the MX record which points to the target network’s mail server, the NS record which points to the host’s name server, the CNAME record which reveals canonical names (aliases) of hosts, the PTR record which maps IP addresses to host names, and the HINFO record which contains host CPU type and operating system. Once the red team has this information, they can use a tool such as Traceroute to further map the topology of the target network. Traceroute utilizes the Internet Control Message Protocol (ICMP) to discover the routers on a path to a target host by sending packets and incrementing the Time To Live (TTL) field in the header of each ICMP packet until the target host is reached. By utilizing Traceroute, red teams are able to determine information about the network topology, trusted routers, and firewall locations.

Once red teams have this information about a target network (either through previous knowledge or footprinting), they can determine the vulnerabilities that exist with the systems and software and attempt to exploit them to gain access to the target network to search for further vulnerabilities.

One advantage that red teams have is that it is relatively easy to gather information about a target network using these techniques. A disadvantage that red teams face is that it is time consuming for red teams to conduct thorough footprints of networks. Another disadvantage of relying solely on red teams to conduct reconnaissance with footprinting is that there is no standardization of the reporting of findings from the tests of different networks throughout the DoD.

Due to the resource constraints that are a reality with DoD red teams, there are multiple benefits to be gained by automating the footprinting process.

2. MAST Methods

While footprinting is not implemented in the current version of the MAST system, it can be developed where the techniques utilized by red teams can be automated (Whois queries, DNS queries, etc.) to determine how vulnerable a particular DoD network is to this sort of information gathering and also to test the network's compliance with firewall configuration policies, etc. The remote scenario execution server can perform this type of policy check prior to an assessment to determine vulnerabilities on various DoD networks and to allow for consistent reporting and feedback. Such feedback allows for better monitoring of network compliance and readiness DoD-wide, as well as trend analysis of common vulnerabilities. The MAST footprinting module could also be used on a scheduled or random basis on various networks DoD-wide, subject to local command and DoD policies, to conduct a "spot check" of compliance of DoD networks outside of organized training events. In this manner, the MAST footprinting module will ensure the IA security readiness posture of DoD networks is within prescribed limits at all times, which will ultimately raise the DoD network security readiness posture on the whole.

Additionally, the MAST system can implement a method for checking individual command's training effectiveness against social engineering phishing attacks. For instance, a module can be created that sends e-mail to junior sailors by using ship's address book for all E-3 and junior sailors and see how many sailors will respond to a request for sensitive information, such as their password, and how long until the query is reported to the network administrators. Metrics to determine the effectiveness of training might include "how many sailors divulged sensitive information?" and "how long did it take for the phishing attempt to be reported to network administrators?" These resultant metrics could be locally stored in the command's scenario execution server's database and also sent back to the initiating scenario generation server (via secure connection) so that the results can be added to the master database. These results could then be compared to past results from that specific command as well as the results from all DoD networks to allow for greater trend analysis, as well as providing consistent data for individual command de-briefings.

B. SCANNING

1. Red Team Methods

Red teams utilize various network scanning techniques to gain a greater understanding of the topology of a target network and to find potential vulnerabilities to exploit. According to the Certified Ethical Hackers Manual, "Scanning refers to a set of procedures for identifying hosts, ports, and services in a network" [6]. The general methodology that red teams employ to scan a target network is to scan for live hosts (i.e., hosts that are online), check for open ports on those hosts, and then scan those hosts for vulnerabilities. The information that red teams gather from these scans allows them to draw a notional network diagram and to focus on vulnerabilities that they intend to attempt to exploit. Red teams utilize a tactic known as a “ping sweep” to determine what hosts on a target network are online. There are various tools that implement this ping sweep functionality for example, Hping is a command line utility that automates the crafting of ICMP Ping packets to determine which hosts are online in a given IP address range. The ping method in relies on the ICMP protocol which can be turned off in IPV4, however, the ping functionality cannot be turned off in IPV6.

Once the red team has determined which hosts are online in a target network, they typically conduct a port scan using a tool such as Hping or Nmap. Port scans rely on the Transmission Control Protocol (TCP), and the “three way handshake” that takes place in order to establish a connection between a server and a client. The three way handshake is executed as follows: a client sends a SYN packet to a specific port on another client, if the port is “open” (has a service listening), the second client responds with a SYN-ACK packet with a sequence number, acknowledging the connection request, to which the initiating client responds with an ACK packet echoing back the sequence number. Once this process is complete, a connection is established between the two clients and they can communicate further. If the port with which a connection attempt is initiated is closed, the client with the closed port will respond with a RST packet. To “tear down” (close) the connection the initiating client responds with a RST packet and the connection is closed. A port scan that executes the full three way handshake is known as a full open scan; however, this type of scan is easily detectable by Intrusion Detection Systems (IDSs) so in an effort to avoid detection, red teams will typically employ a form of “stealth” scanning.

Stealth scanning allows red teams the ability to determine which ports are open on target hosts while bypassing firewall rules and logging mechanisms to disguise their traffic as usual network traffic. An example of a stealth scan is known as a TCP half-open scan. A half-open scan is the same as a full open scan with the exception that the client initiating the connection sends a RST packet once it receives the initial SYN-ACK packet from the target host and, by doing so, the initiating client closes the connection initiation before a connection is ever established. The tools that red teams use to conduct these scans can be configured to avoid detection by executing stealth scans and only scanning well-known ports (port 25 SMTP, port 80 Web server, etc.).

An advantage that red teams have with respect to scanning target networks is that they have a myriad of tools and techniques to conduct scanning. Another advantage that red teams have is that these tools are extremely effective at gathering data on a target network in short amounts of time.

One disadvantage of using red teams to conduct this type of scanning and information gathering for network security training is that every training event is different. Depending on the red team personnel conducting the scanning and information gathering for a particular training exercise, they utilize different tools and methods to gather information from different avenues. This is due in part to red team purpose: to identify and exploit weaknesses to identify security issues, not to provide administrator training. Due to this variability in scanning methodologies it is extremely difficult to gather consistent feedback from one training event to the next. Another disadvantage that red teams face with this type of scanning is that the information that they are able to gather is limited until they have access beyond the De-Militarized Zone (DMZ) or beyond the firewall, of the network. However, once the red team is able to find a vulnerability to exploit that allows them to gain access to the target network, they are able to employ these or other techniques to gather more information about the target network.

2. MAST Methods

MAST scanning modules can implement the various techniques and functionality of red team's tools and will allow for thorough scanning of networks while not increasing the risk to the networks. The MAST system allows for the testing of firewall policies, and network administrators' knowledge of Pre-Planned Responses (PPRs) while ensuring that the test scans are consistent and repeatable on all networks DoD-wide. Due to the consistent and repeatable features implemented by the MAST system scanning modules, it will be much easier to replicate training and consolidate consistent feedback from the results of scans, which allows for trend analysis of DoD networks so that we can identify trends and better shape our PPRs to adversary’s scanning techniques, which will ultimately result in improved overall IA posture of DoD networks.

An advantage of conducting training with the MAST system for network administrators and operators to recognize scanning behaviors is that the training is repeatable and can be reproduced once the network administrators have addressed previously detected vulnerabilities. The feedback from the MAST system will reflect the improved IA posture of the trainee thus providing more timely feedback of the defensive IA posture of the network.

A disadvantage of MAST system is that as new vulnerabilities are discovered, they would require new modules to be written for the MAST system. However, once the new modules are written, the red teams or other training entities will be able to conduct training with the new module on all of the various DoD networks, allowing for a quick turnaround on training specific to new vulnerabilities and their associated PPRs, thus increasing the DoD’s IA posture to emerging threats in a more timely manner.

C. GAINING ACCESS

1. Red Team Methods

Red teams use various techniques, such as password eavesdropping, brute force, or dictionary password-cracking attacks, to gain access to a target system which they can then use to escalate privileges on that system, run exploits, etc. Additionally, there are a vast number of exploitable vulnerabilities that red teams use to gain access to a network. Once the red team has sufficient information about the operating systems and software deployed on a given target network, they are able to determine which vulnerabilities to attempt to exploit to gain access to the network. One method red teams use to gain access to a network is to determine the level of patching of particular systems on the network and exploit known vulnerabilities that have not been patched. Red teams are successful using this approach due to the myriad of exploitable software bugs that exist in current software. According to DARPA, for every one thousand lines of code in software, there are one to five bugs introduced [18] and since modern operating systems and security software size is on the order of millions of lines of code, and the fact that we are constantly implementing new software, the attack surface for red teams (as well as adversaries), is extremely large.

An example of this type of exploitable vulnerability is the Microsoft Windows Remote Procedure Call (RPC) Distributed Component Object Model (DCOM). The RPC-DCOM, if left unpatched in various Windows operating systems, is vulnerable to a buffer overflow attack which allows the attacker to run arbitrary malicious code on the target system with local system privileges [19].

Once the red team finds an unpatched vulnerability they can execute their own malicious code to manipulate the target system and return a command prompt with system privileges, for instance. Once the red team has unabated access to a system on a network, they are able to utilize their various other methods in order to maintain access to the compromised system, gather further information about other systems on the network, and a multitude of other nefarious activities.

An advantage that red teams have when attempting to gain access to DoD networks is that a vast majority of DoD networks have similar software loads, that is they are running the same software. Knowing this, a red team is able to attempt to exploit known vulnerabilities that they have had success exploiting in the past.

As discussed with Dave Aland, a disadvantage of red teams with respect to gaining access to a target network is that the feedback that is provided to the trainee network administrators typically focuses only on the exploits that the red team used successfully [20]. There is value to the network administrators in knowing what exploits the red team attempted unsuccessfully, and this type of feedback also more accurately portrays the overall security posture of the network.

2. MAST Methods

The MAST system will be preinstalled on DoD networks to facilitate frequent and consistent training. As a result of this, the MAST system will technically already have access to a given trainee network. The advantage that the MAST system has over red teams with respect to gaining access is that it does not require any malicious scripts or files in order to gain access. Additionally, the MAST system does not need to engage in other nefarious activities such as password cracking in order to simulate malicious activity. Due to the fact that the MAST system is a “trusted” system, it can facilitate effective training without introducing any malware to the trainee network thereby increasing training value without a concomitant increase in risk.

D. MAINTAINING ACCESS

1. Red Team Methods

Red teams, and especially adversaries, typically utilize a “backdoor” or possibly a remote access Trojan Horse to maintain access to a compromised system. A backdoor is code that is used to secure remote access to a compromised system and effectively bypass normal authentication mechanisms (i.e., user name and password). A Trojan Horse is “a program in which the malicious or harmful code is contained inside apparently harmless programming or data in such a way that it can get control and cause damage” [6]. An example of a remote access Trojan Horse is an executable file, bound to an apparently benign file such as a Microsoft Word .doc file, that installs a Netcat server on the target machine, which allows the attacker to connect remotely to the target machine via a command shell by opening an unused port of the attacker’s choice and sending commands to the target machine. The red team or attacker has then set up a mechanism for maintaining access to the compromised system and is able to reconnect at will to send commands to the compromised system. There are a myriad of techniques by which a red team can set up unlimited remote access to a compromised machine. Conversely, a red team may just continue to use whatever exploit that allowed them access in the first place since most training exercises are not longer than two weeks and, as long as they are undetected, the chance of system administrators discovering and correcting the vulnerability in that time is low.

An advantage that the red teams have when training and assessing a particular network is that they have a multitude of exploits and tools to maintain access once they have compromised a system. A disadvantage that red teams have is that they may be limited on what exploits and tools that they can use due to the concomitant risk that they induce.

2. MAST Methods

As discussed previously, the MAST system is a trusted system that only mimics malicious software activity and, therefore, the MAST system has the required "access" to conduct training. This implicit access allows for training to be conducted on the trainee network without a concomitant increase in risk.

The biggest advantage that the MAST system has with respect to maintaining access is that there is no need for nefarious techniques to be utilized in order to maintain access to the trainee network. As previously mentioned, the MAST system allows for training to be conducted without an increase in risk to the network.

E. COVERING TRACKS

1. Red Team Methods

As discussed in Chapter II and depending on the training objectives of a particular training session, red teams may limit their effort on covering their tracks to allow for the trainee to determine what vulnerabilities were exploited. However, a common method employed by hackers and red teams alike is to utilize a “rootkit” to cover their tracks. According to *Gray Hat Hacking the Ethical Hacker’s Handbook*, a rootkit is “software that hides itself and other software from system administrators in order to perform some nefarious task” [21]. By using a rootkit, the red team is able to hide malicious files and applications from system administrators. Additionally, red teams may also cover their tracks by deleting log entries in Intrusion Detection Systems (IDS) and the attacked systems.

An advantage that red teams have in covering their tracks is that they have a myriad of tools and techniques to help avoid detection by IDS and system administrators. Conversely, however, red teams may be limited in how many of their tools and techniques that they may employ so that the trainee network administrators are given the opportunity to identify the source of attacks on their network.

2. MAST Methods

The MAST system provides training without having to cover tracks since no actual malicious software is installed on any system.

An advantage of the MAST system is the ability to conduct IA security training for system administrators, DoD-wide, without having to utilize some of the nefarious methods that red teams employ to cover their tracks.

F. FURTHER MAST COMPARISON WITH CURRENT TRAINING METHODS

In order to determine how the MAST system as a training tool compares with other training methods and to verify and validate the MAST system as a training tool, we define some metrics from which to base our comparison. With these metrics defined, we will then compare the MAST system with other network administrator training methods, specifically the Rapid Experience Builder (RaD-X), red teams, and the Metasploit Framework.

RaD-X is a training tool for network administrators that delivers “hands-on” training with malware [22], [23]. RaD-X is a deployable training network which allows for training to be conducted in an isolated environment.

The Metasploit Framework is an open source tool for penetration testing and network security auditing [6], [24]. The Metasploit Framework has roughly three hundred exploits for gaining access to various target systems (Windows, Unix, etc.). The exploits allow the user to run various payloads of code on the target system and determine which vulnerabilities are exploitable on a given system. Metasploit can be used to train network administrators on recognizing malicious software on their network as well as for penetration testing.

For the metrics defined below, we have assigned scores ranging from low to high as a basis for comparison of the various training methods and to facilitate further discussion of the advantages and disadvantages of each training method. The costs discussed in this section are summarized in Table 1.

1. Holistic Cost of Training Methods

The holistic cost of a system is the overall cost, including personnel, equipment, travel, etc., associated with conducting training for each method. The cost of conducting training with the MAST system is medium-low since the training is software based and the bulk of the cost will be the deployment of the software across the DoD with the additional cost of training system administrators on the MAST system. These costs will be outweighed by the resultant increase in IA posture throughout the DoD and if we prevent just one catastrophic malware attack as a result of better trained system administrators, then the MAST system proved its worth.

RaD-X is a self-contained network of twenty to twenty five workstations that allows for training in a "sandbox" environment; that is, the training network is completely isolated. The cost of deploying RaD-X is medium-high due to the fact that the whole system must be shipped to the trainee, along with the temporary duty (TDY) costs of the system operators, training of operators, etc.

The cost of conducting training with red teams is high due to the cost of training and equipping red teams, as well as the costs for research and development for red teams to continuously keep pace with currently evolving threats. The cost associated with research and development to keep pace with evolving threats applies to all training methods but is higher with red teams since the newfound methods will have to be trained to for various red team personnel. Additionally, due to the increase in demand for red teams as the cyber warfare area garners increasing attention, the costs for red teams to meet the increase in demand will continue to rise.

The cost to conduct training with a tool such as the Metasploit Framework is relatively low. The majority of the cost incurred if we used Metasploit as a tool for system administrators would be as a result of training personnel to use Metasploit effectively.

	Training Method Comparison				
	Training Attributes	**MAST**	**RaD-X**	**Red Team**	**Metasploit**
1	**Holistic Cost**	Med-Low	Med-High	High	Low
2	**Speed to Product**	Med-Low	High (fast)	Medium	Medium
3	**Coverage of Users**	High	Low	High	Low
	Coverage of Exploit Domain	Medium	High	Medium	High
4	**Risk**	Med-Low	Low	Med-High	Med-High
5	**Realism of Attack Vector**	Medium	High	Med-High	High
	Realism of Training Environment	Med-High	Med-Low	High	High
6	**Training Auditing (feedback)**	High	High	Med-Low	Med-Low
7	**Training Availability (i.e. frequency)**	High	Low	Low	Medium
8	**Consistency of Training (V+V)**	High	High	Low	Med-Low
9	**Ease of use of Training Tool**	High	Low	Low	Low
10	**Training Infrastructure**	Distributed	Centralized	Centralized	Centralized

Table 1. Comparison of Training Tools.

2. Speed to Product

"Speed to product" is how long it takes after a new piece of malicious software or a new exploit is discovered for a particular training tool to incorporate the newfound malware/exploit in training scenarios.

The speed to product of the MAST system is medium-low due to the fact that once a new piece of malware or exploit is discovered, it has to be analyzed to determine its attributes and then a new module written for the MAST system to mimic these attributes.

The speed to product of training on new malware or a new exploit with RaD-X is high since the new malware can be released to the trainer without the trainer having to analyze the malware in detail.

The speed to product of training on new malware or a new exploit with red teams is medium due to the fact that red teams must analyze the new malware or exploit and determine exactly how it operates to utilize the techniques employed by the newfound malware or exploit.

The speed to product of new malware or exploits with the Metasploit Framework is medium since the framework developers must analyze the new malware or exploit and then implement it for the framework.

3. Coverage of Users and Exploit Domain

The "coverage of users" for training conducted with the MAST system is high since the training is conducted on the actual network of the DoD trainee entity. An example of the high level of user coverage of a MAST training event is the "phishing" e-mail module to test the training effectiveness of the trainee’s command by sending a phishing message to the entire command and logging how many users clicked the nefarious link. The “coverage of the exploit domain” for the MAST system is medium. This is due to the fact that modules will not necessarily be developed for every piece of malware since a lot of malware uses similar techniques (i.e., scanning behaviors of various worms).

The coverage of users for RaD-X is low since the system has to be transported to the trainee command and training is typically only conducted on senior network administrators and operators; RaD-X is not intended to train the average user on malware. However, the coverage of the exploit domain with RaD-X is high since any malware can be run on it safely due to the isolated nature of the training RaD-X provides.

The coverage of users with red teams is high since, like the MAST system, the training is conducted on the actual DoD network. The coverage of the exploit domain is medium with red teams due to the fact that they cannot use some of their more nefarious exploits due to the increase in risk to the trainee’s networks.

With Metasploit, the coverage of users is medium due to the fact that Metasploit exploits are targeted at particular systems and to cover a large number of users or systems on a network would require a lot of time and repetition of work. The coverage of the exploit domain with Metasploit is high since there are over three hundred exploits built in to the Metasploit Framework.

4. Risk Associated with Training Tool

The risk associated with conducting training with the MAST system is medium-low due to the fact that no actual malware is ever used on the trainee network. However, since the MAST modules will exhibit the behaviors of malware, there is some inherent risk associated with increased network traffic causing latency on the network as well as the HBSS intrusion detection systems potentially blocking legitimate network traffic from a host exhibiting malicious behaviors.

RaD-X provides low risk training because the training is conducted in an isolated sandbox network and no malware is ever used on an actual DoD network.

The risk associated with training conducted by red teams is medium primarily due to the safeguards and limitations placed on red teams to protect the trainee network.

With Metasploit, the risk to the trainee network is high because of the nefarious methods which are used to exploit vulnerabilities. Safeguards would have to be implemented to use Metasploit to conduct network administrator training to ensure that risk is limited to acceptable levels.

5. Realism of Attack Vector and Training Environment

When conducting training with the MAST system, realism of the attack vector is medium since the actual malware is mimicked and no actual malware is used in the training. However, the realism of the training environment with the

MAST system is high due to the fact that the training is conducted on the actual network of the trainee command.

With RaD-X, the realism of the attack vector is high due to the fact that actual malware is used for training. The realism of the training environment is medium-low with RaD-X because the training is conducted on an isolated training network and not the actual network that the administrators oversee on a day-to-day basis.

When red teams conduct training, the realism of the attack vectors is medium-high and is only limited by safeguards and constraints put in place to protect the trainee network. The realism of the training environment is high with red teams since they are also conducting training using actual exploits on the actual network that the administrators oversee.

With Metasploit, the realism of the attack vectors is high since the nefarious attack vectors are built into the framework. Additionally, the realism of the training environment is high with Metasploit as long as the training is conducted on the actual DoD network.

6. Training Auditing

The "training auditing" (feedback) of training conducted with the MAST system is high since the results of each training event are logged in the MAST database. Additionally, due to the consistent and repeatable nature of training conducted with the MAST system, the feedback is consistent across all training events conducted DoD-wide.

The training auditing with RaD-X is high, also, since feedback is provided to the students after each training session.

With red teams, the training auditing is medium. This is due to a couple of factors. First, red teams typically provide feedback only on the exploits that they successfully executed, however; there is value to the trainee in knowing what exploits the red team attempted unsuccessfully. Second, since different red teams use different attack vectors and methods, the feedback from one training event to the next is not standardized.

Conducting training with Metasploit would encounter the same feedback issues that red teams face since it would be up to the trainer to provide feedback on the exploits used and would therefore be subject to different methods from training event to training event with the resulting inconsistency of feedback.

7. Training Availability

The "training availability" or frequency of training with the MAST system is high due to the fact that system administrators or trusted agents within individual commands (CSTT for instance), can conduct training on a monthly or more frequent basis. The training objectives could be incorporated in a ship's quarterly training plan and addressed accordingly.

With RaD-X, the training availability is low since RaD-X is a limited asset and individual commands may only have occasional access to it.

As previously discussed, red teams are also a limited asset and the demand for their services is ever increasing. Due to this fact, the training availability of red teams is considered low.

With Metasploit, the training availability is medium due to the fact that it could only be used to train a limited number of people since for each trainer it would be a one-to-one mapping of trainer-to-trainee for each training event. The other training methods enable a single trainer to train many trainees simultaneously.

8. Consistency of Training

The "consistency of training" with the MAST system is high since the same module (a worm propagating, for instance) could be repeated on various networks throughout the DoD. With the MAST system, each training event can be tailored to meet desired training objectives and the individual modules that are executed to meet these training objectives will exhibit the same signatures and behaviors every time, which provides consistent feedback and training.

With RaD-X, the consistency of training is also high because the training is also repeatable.

Consistency of training with red teams is low due to the variability of exploits used and methods employed by various red teams.

Metasploit offers medium consistency of training since it offers various exploit methods and payloads of malicious code. Similarly to red teams, training with Metasploit would be variable due to the vast number of exploits and payloads.

9. Ease of Use

The MAST system has a high “ease of use” for training because of its modular design of functionality. The entity conducting the training on a particular trainee network will pick the modules necessary to fulfill the training objectives for the given training event. Additionally, the collection of data from each training event is automatically reported which makes the MAST system easier to use for trainers.

Rad-X has a low ease of use as the network must be shipped to the trainee location, set up, and tested prior to conducting training.

The ease of use associated with red teams conducting training is low due to the fact that the training events and feedback associated with them are not standardized.

The ease of use of Metasploit as a training tool is since the Metasploit Framework does not support distributed training of many clients simultaneously in the manner that the MAST system does.

10. Training Infrastructure

The training infrastructure for the MAST system is distributed and takes advantage of the client-server architecture of networks. The capability that is provided with the remote scenario generation server allows for multiple training scenarios to be conducted on multiple trainee units remotely. Additionally, the scenario execution servers (local to each network) provide local training capability to each command.

With RaD-X, the training infrastructure is centralized as the training network is an isolated sandbox and training can only be conducted locally on that network.

The training infrastructure with red teams is centralized since their training is deployed from a central location. Red teams, however, are capable of conducting training on multiple units simultaneously.

The training infrastructure with Metasploit is distributed due to the fact that training can be conducted independently on various remote networks as well as locally.

As a result of the comparisons made between the training tools in this section, we posit that the MAST system is indeed a viable solution to the increased training demand in the cyber warfare domain. The MAST system will not replace red teams since their skill sets are of critical importance to keep up with the constantly evolving threats in the cyber domain. However, the MAST system addresses the critical need for more frequent and consistent training and will augment the training currently provided by red teams while easing the burden on that limited resource.

G. STRATEGY TO MEASURE EFFECTIVENESS OF MAST

To measure the effectiveness of the MAST system, we wish to verify and validate the MAST system software, that is prove that the system is doing the right job according to specifications (validation), and that the software is doing the job correctly (verification). To measure the effectiveness of the MAST system as a training tool and to verify and validate the software, we propose the following testing techniques.

1. Network Traffic Analysis

One strategy to measure the effectiveness of the MAST system as a training tool is to compare network traffic generated from individual MAST modules and compare the traffic to known traffic signatures of the mimicked malware. The modules of the MAST system mimic well known malicious behaviors of particular attacks (for instance, a worm propagating). Since we are mimicking well-known malware activity it will be fairly straight-forward to determine whether or not our module is accurately portraying the behavior of a particular piece of malware. A strategy for measuring the traffic might be to utilize a packet-capturing program such as Wireshark or TCP-Dump to capture traffic passing through a switch on the network so that we are capturing all traffic on the network segment. Once we are capturing traffic on the network, we would run the particular MAST module that we intend to test and upon completion of the execution of the module, analyze the traffic, comparing it with the known signatures to determine how accurately we mimicked the actual malware. We would then be able to determine if modifications to a particular module are necessary based on the results of this comparison and adjust the behavior of our module accordingly.

2. Intrusion Detection Systems

Another strategy to measure the effectiveness of the MAST system modules is to run the modules on our virtual CG-71 implementation platform with HBSS. Upon completion of the execution of a given module, we would inspect the logs of the Host Intrusion Prevention System (HIPS) and the firewall to ensure that the expected log entries are generated as a result of our simulated malware. Additionally, we would monitor the alarms and log entries that are generated by the ePolicy Orchestrator in HBSS to ensure that our module is "tripping" the appropriate alarms.

3. Live Testing

Once we are confident that the MAST system accurately mimics the malware we have implemented in the modules, we can test the MAST system's scalability on an IA "range." The Defense Department Information Assurance Range is a sandbox environment that simulates the Global Information Grid and is operated by DISA and the United States Marine Corp [25]. The IA range would enable us to test the MAST system on a large-scale implementation and to further verify and validate the MAST system as a viable training tool.

H. SUMMARY

In this chapter, we discussed the techniques that red teams and ethical hackers use to advance an attack on a trainee network. We also discussed how the MAST system accomplishes these same tasks and discussed advantages and disadvantages of the red teams methods as well as the corresponding MAST methods. Additionally, we then defined some metrics by which to compare the MAST system with other training tools and discussed the strengths and weaknesses of each tool. From this comparison, we assert that the MAST system is indeed a viable solution to some of the constraints that the DoD is faced with while relying heavily on red teams to conduct cyber security training. Finally, we proposed some methods for measuring the effectiveness of the MAST system as a viable training tool. In the next chapter we discuss conclusions from this thesis and outline a way ahead on the project with future work.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND FUTURE WORK

A. CONCLUSIONS

In this thesis, we performed a verification and validation analysis of the suitability of the MALWARE Mimic concept of the MAST system as a methodology for conducting network administrator network security training and awareness, alleviation of red team availability constraints, and network user security awareness training. We also proposed a strategy by which the effectiveness of the MAST system for increasing such network security awareness and elevating the information assurance posture of distributed command networks can be measured.

In Chapter III, we discussed the MAST system architecture and features that it implements. Most notably, we discussed the safety features implemented in the MAST system that enable us to conduct software based network IA security training. These safety features ensure that the MAST system will enable the DoD to conduct training without a concomitant increase in risk to network resources. Additionally, we discussed the implementation platform that we have constructed to simulate the hardware and software of a shipboard network. The implementation platform enables us to thoroughly test the MAST system with the Host Based Security System currently deployed on DoD networks to ensure that the MAST system provides the most realistic training possible.

As discussed in Chapter IV, the MAST system is not intended to replace red teams, since their skills will always be necessary in order to discover new vulnerabilities and create safeguards to them, as well as thoroughly test a network’s IA posture. The MAST system will augment the assessment capability provided by red teams and allow for better IA awareness and an increase in overall IA posture DoD-wide.

Additionally, in Chapter IV, we defined various metrics to compare the MAST system with other network administrator training methods specifically, RaD-X, and Metasploit, as well as red teams. We discussed the advantages and disadvantages of each training method and assigned each method a score accordingly. As a result of this analysis, we assert that the MAST system is indeed a viable solution to decrease the DoD’s dependence on red teams to conduct network IA training. The MAST system also enables the DoD to gather more consistent feedback from individual training events that, in turn, facilitates trend analysis of vulnerabilities in DoD networks. It is reasonable to conclude that the increased frequency and consistency of training events facilitated with the MAST system will pay huge dividends in DoD network security.

We demonstrated that the MAST system is a viable training method and that it will ensure that more frequent and consistent training is conducted with DoD network administrators thereby increasing the overall IA security posture on the whole.

B. FUTURE WORK

1. More Advanced Modules

The training value of the current iteration of the MAST system has significantly improved over the previous version of the MALWARE Mimic software. However, to fully realize the MAST system’s training potential, more advanced modules will need to be created.

As discussed in Chapter II, we foresee a module that more thoroughly implements the behaviors of a worm propagating on the network. The worm propagation module could exhibit more distinct signatures in addition to “scanning” for vulnerable hosts, such as simulating replicating itself on further “infected” hosts that will trigger responses from the HBSS Host Intrusion Prevention System and elicit appropriate responses from trainees. Furthermore, the MAST system modules could implement various iterations of the worm module to exhibit the different signatures from various worms to provide training on the various methods which worms use to propagate and the behaviors and signatures that they exhibit.

Furthermore, as discussed in Chapter II, we foresee a module that more robustly implements virus behaviors. The virus module could be “sent” to unsuspecting users as an e-mail attachment and will “spread” depending on how many users click the nefarious link. Consistent with the methodology and purpose of the MAST, such modules will not actually infect the hosts, but rather trigger indicators through prepositioned agents to mimic the infection. Similarly, as previously discussed, various virus modules could be implemented to exhibit the behaviors and signatures of different viruses to broaden the training to cover more of the exploit domain.

The footprinting module that was discussed in detail in Chapter IV also requires implementation. This module will gather information on the IA state of the trainee network and report potential vulnerabilities. The footprinting module could also potentially interact with other modules to shape how they propagate or possibly give feedback to the trainer on what modules to use to provide the most realistic training based on the IA posture of the network.

2. Standardized Feedback Criteria

To maximize the training value of the MAST system and leverage the consistency and repeatability of training conducted, a thorough analysis of the feedback that commanders, trainers, and trainees require should be undertaken. An exhaustive list of feedback requirements should be compiled for each type of malicious software mimicked and then those requirements can be implemented in the database and Mast software to ensure that the required feedback is provided by each MAST module. This ensures that the thorough and consistent feedback that is desired of the MAST system will be implemented and thereby greatly impact the quality of feedback from IA training events through standardization.

3. Security Implications of MAST

The MAST software is still in the implementation phase as it goes through cyclic development. With each iteration of the software, it is prudent to conduct a security assessment of the software and ensure that the MAST system does not introduce new vulnerabilities to DoD networks. As discussed in Chapter III, it is crucial that the communications between the remote scenario generation server and each trainee command’s local scenario execution servers for module injection and training feedback data are encrypted and secure. The securing of the communication channels of the MAST system will prevent adversaries from gaining access to the system and attempting to exploit it. Additionally, the MAST software should be thoroughly examined to determine if bugs exist and discovered bugs fixed to ensure the continued security of the software.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] W. R. Taff Jr. and P. M. Salevski, "Malware mimics for network security assessment." M.S. thesis, Naval Postgraduate School, Mar 2011.

[2] J. F. Sandoz, "Red teaming: A means to military transformation," Final IDA Paper P-3580, Jan. 2001. Available from DTIC, Alexandria, VA.

[3] "National Information Assurance (IA) Glossary." Committee on National Security Systems, 26 Apr 2010.

[4] R. Gile, "Global War Game Second Series 1984-1988." Newport, RI: Naval War College Press, 1998.

[5] "2010 Press Release - U.S. Naval Academy Triumphs in 10th Annual Cyber Defense Exercise," 26 Apr 2010. [Online]. Available: http://www.nsa.gov/public_info/press_room/2010/cdx.shtml. Accessed 20-Oct-2011.

[6] S. Cote, R. Petrunic, P. Branka, N. T. Khalil, M. Schumak, C. Chavez, and A. Silva, *Certified Ethical Hacker: Ethical Hacking and Countermeasures, Courseware Guide v7.1*, vol. 1. Albuquerque, NM: EC-Council USA, 2011.

[7] D. J. Aland, "Towards Better Control of Information Assurance Assessments in Exercise Settings," Wyle Research Labs, Arlington, Virginia, 2008.

[8] T. Nash, "An Undirected Attack Against Critical Infrastructure A Case Study for Improving Your Control System Security." U.S. Department of Homeland Security Vulnerability & Risk Assessment Program (VRAP), Sep 2005.

[9] G. Gu, M. Sharif, X. Qin, D. Dagon, W. Lee, and G. Riley, "Worm detection, early warning and response based on local victim information," in *Computer Security Applications Conference*, 2004,pp. 136-145.

[10] E. Messmer, “The botnet world is booming,” 09 Jul 2009. [Online]. Available: http://www.networkworld.com/news/2009/070909-botnets-increasing.html?page=1. Accessed 20 Jan 2012.

[11] P. Szor, *The Art of Computer Virus Research and Defense*, 1st ed. Upper Saddle River, NJ: Addison Wesley, 2005.

[12] “I Love You Virus Details and Removal.” [Online]. Available: http://www.fireav.com/virusinfo/library/love.htm. Accessed 20 Jan 2012.

[13] “Iloveyou,” *Wikipedia* [Online]. Available: http://en.wikipedia.org/wiki/ILOVEYOU. Accessed 20 Jan 2012.

[14] “Virtual Machines, Virtual Server, Virtual Infrastructure,” *Virtualization Basics*. [Online]. Available: http://www.vmware.com/virtualization/virtual-machine.html. Accessed 31 Jan 2012.

[15] “Resource Management in VMware ESX Server 3.” [Online]. Available: http://download3.vmware.com/vmworld/2006/tac9726.pdf. Accessed 01 Feb 2012.

[16] *CND-OSE 1.2 Host Based Security System Training Handbook*. San Diego, CA: ManTech, 2011

[17] “Strategic Command Directive 527-1_27JAN2006 Information Operations Condition-INFOCON-System.pdf,” 27-Jan-2006. [Online]. Available: http://info.publicintelligence.net/StrategicCommandDirective527-1_27JAN2006InformationOperationsCondition-INFOCON-System.pdf. Accessed 10 Feb 2012.

[18] D. Ragsdale, “DARPA’s Cyber Analytical Framework: Opportunities for Cyber Researchers.” [Online]. Available: http://www.cs.tamu.edu/tref/ragsdale. Accessed: 17 Jan 2012.

[19] “Microsoft Security Bulletin MS03-026: Buffer Overrun In RPC Interface Could Allow Code Execution (823980),” 16-Jul-2003. [Online]. Available: http://technet.microsoft.com/en-us/security/bulletin/ms03-026. Accessed: 2 Mar 2012.

[20] D.J. Aland, private communication. 29 Feb 2012, Monterey, California.

[21] A. Harper, J. Ness, G. Lenkey, S. Harris, C. Eagle, and T. Williams, *Gray Hat Hacking The Ethical Hacker’s Handbook*, Third ed. New York: McGraw-Hill, 2011.

[22] “IA_DoD_IA_Training_Products.pdf.” [Online]. Available: http://www.disa.mil/News/Conferences-and-Events/~/media/Files/DISA/News/Conference/CIF/Briefing/IA_DoD_IA_Training_Products.pdf. Accessed: 15 Mar 2012.

[23] “March15_FISSEA-isslob-awareness-tier1-tier2-panel-GBieber.pdf.” [Online]. Available: http://csrc.nist.gov/organizations/fissea/2011-conference/presentations/March15_FISSEA-isslob-awareness-tier1-tier2-panel-GBieber.pdf. Accessed: 14 Feb 2012.

[24] “Learn More | Metasploit Project.” [Online]. Available: http://www.metasploit.com/about/. Accessed: 3 Mar 2012.

[25] H. Kenyon, “DoD Cyber Range simulates Global Information Grid — Defense Systems,” 02-Dec-2010. [Online]. Available: http://defensesystems.com/articles/2010/12/01/dod-launches-cyber-test-range.aspx. Accessed: 5 Mar 2012.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Captain David Aland, USN (Ret.)
   Office of the Director, Operational Test & Evaluation
   Washington, D.C.

4. Dr. Gurminder Singh
   Naval Postgraduate School
   Monterey, California

5. Mr. John H. Gibson
   Naval Postgraduate School
   Monterey, California